NAVAL POSTGRADUATE SCHOOL
Monterey, California

DISSERTATION

JOINT DEMODULATION OF LOW-ENTROPY NARROWBAND COCHANNEL SIGNALS

by

Timothy J. Meehan

December 2006

Dissertation Supervisor: Frank E. Kragh

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE December 2006

3. REPORT TYPE AND DATES COVERED Doctoral Dissertation

4. TITLE AND SUBTITLE Joint Demodulation of Low-Entropy Narrowband Cochannel Signals

5. FUNDING NUMBERS

6. AUTHORS Meehan, Timothy J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT(*maximum 200 words*)

Reception of one or more signals, overlapping in frequency and time with the desired signal, is commonly called cochannel interference. Joint detection is the optimal minimum probability of error decision rule for cochannel interference. This dissertation investigates the optimum approach and a number of suboptimum approaches to joint detection when *a priori* information based in fields, or sets of transmitted symbols, is available. In the general case the solution presents itself as a time-varying estimation problem that may be efficiently solved with a modified Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm.
The low-entropy properties of a particular signal of interest, the Automatic Identification System (AIS), are presented. Prediction methods are developed for this signal to be used as *a priori* information for a joint field-based maximum *a posteriori* (MAP) detector. Advanced joint detection techniques to mitigate cochannel interference are found to have superior bit error rate (BER) performance than can be obtained compared to traditional methods.

14. SUBJECT TERMS Multiuser Detection, Signal Processing, Communications, Low-Entropy, Classification, Joint Detection

15. NUMBER OF PAGES 206

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified

20. LIMITATION OF ABSTRACT UL

THIS PAGE INTENTIONALLY LEFT BLANK

JOINT DEMODULATION OF LOW-ENTROPY NARROW BAND COCHANNEL SIGNALS

Timothy J. Meehan
Civilian , United States Navy
B.S.E.E., University of Maryland , 1996
M.S.E.E., Johns Hopkins University, 2000

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2006

Author: Timothy J. Meehan

Approved by: Frank E. Kragh
Assistant Professor of Electrical
and Computer Engineering
Dissertation Supervisor

Herschel H. Loomis
Professor of Electrical
and Computer Engineering
Committee Chair

Alan A. Ross
Visiting Professor Space
Systems Academic Group

Charles W. Therrien
Professor of Electrical and
Computer Engineering

Kenneth A. Clark
External Member NRL

Approved by: Jeffrey B. Knorr, Chair, Department of Electrical and
Computer Engineering

Approved by: Julie Filizetti, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Reception of one or more signals, overlapping in frequency and time with the desired signal, is commonly called cochannel interference. Joint detection is the optimal minimum probability of error decision rule for cochannel interference. This dissertation investigates the optimum approach and a number of suboptimum approaches to joint detection when *a priori* information based in fields, or sets of transmitted symbols, is available. In the general case the solution presents itself as a time-varying estimation problem that may be efficiently solved with a modified Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm.

The low-entropy properties of a particular signal of interest, the Automatic Identification System (AIS), are presented. Prediction methods are developed for this signal to be used as *a priori* information for a joint field-based maximum *a posteriori* (MAP) detector. Advanced joint detection techniques to mitigate cochannel interference are found to have superior bit error rate (BER) performance than can be obtained compared to traditional methods.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

LIST OF ACRONYMS

AIS	Automatic Identification System
AWGN	additive white Gaussian noise
BCJR	Bahl, Cocke, Jelinek and Raviv
BER	bit error rate
BMLSD	Blind Maximum Likelihood Sequence Detection
CCI	cochannel interference
CDMA	code division multiple access
CM-SAIC	Constant Modulus - Single-Antenna Interference Cancellation
CPM	continuous-phase modulation
CRB	Cramer-Rao bound
CRC	cyclic redundancy check
DFT	discrete Fourier transform
FCS	frame check sequence
FER	field error rate
FFT	fast Fourier transform
FIR	finite impulse response
FPGA	Field Programmable Gate Array
FSK	frequency shift keying
GMSK	Gaussian minimum shift keying
GPP	general purpose processor

GSM	Global System for Mobile Communications
HDLC	High-level Data Link Control
IF	intermediate frequency
i.i.d.	independent identically distributed
IMO	International Maritime Organization
ISI	intersymbol interference
ITU	International Telecommunication Union
JFBMAPD	joint field-based maximum *a posteriori* detector
JMAPSD	joint maximum *a posteriori* sequence detector
JMLSD	joint maximum likelihood sequence detection
JMLSE	joint maximum likelihood sequence estimation
LEO	low earth orbit
LOS	line-of-sight
LPF	low-pass filter
LRIT	Long Range Identification and Tracking
MAC	multiply and accumulate
MAI	multiple access interference
MAP	maximum *a posteriori*
MAPSD	maximum *a posteriori* sequence detector
MLE	maximum likelihood estimate
ML	maximum likelihood
MLSD	maximum likelihood sequence detection

MLSE	maximum-likelihood sequence estimation
MMSE	minimum mean-square error
MMSI	Maritime Mobile Service Identity
MSE	mean squared error
MSK	minimum-shift keying
NRZI	non-return-to-zero inverted
NRZ	non-return-to-zero
OQPSK	offset quadrature phase shift keying
PAM	pulse-amplitude modulated
PDF	probability density function
PMF	probability mass function
PSP	Per-Survivor Processing
QPSK	quadrature phase shift keying
RFID	radio frequency identification
RLL	Run Length Limited
SAIC	Single-Antenna Interference Cancellation
SAR	synthetic aperture radar
SNR	signal-to-noise ratio
SOLAS	safety of life at sea
SOTDMA	self organized time division multiple access
TDMA	time division multiple access
UAV	unmanned aerial vehicle

USCG	United States Coast Guard
UTC	universal time
VHF	very high frequency
WSS	wide-sense stationary

ACKNOWLEDGEMENTS

The completion of my doctoral program is the result of contributions from many individuals. I would like to like to express my most heartfelt gratitude to all my friends, family, and colleagues who have supported me along the way.

I am very grateful for the love and support provided by my wife Mindy. With great patience she has provided encouragement. She had no idea what she was in for when she hopped on her bicycle to join me on this wonderful journey.

Many thanks are owed to those at the NRL who helped me get where I am. Special thanks to Charmaine Gilbreath and Anne Reed who brought me out of Ohio to work at the NRL. Kenny Clark who got me interested in FPGA design, inspired me to try new things, and encouraged me to continue my education.

James Hicks has been a valuable resource, helping me narrow down my topic, and acting as a sounding board for new ideas.

My dissertation committee have been a pleasure to work with. My advisor Prof. Frank Kragh has provided more time and support than I could have ever asked for. All of my committee members – Prof. Herschel Loomis, Prof. Frank Kragh, Prof. Alan Ross, Prof. Charles Therrien and Dr. Kenny Clark – provided much needed direction and assistance.

I would also like to thank all of the students and faculty at NPS who helped me along the way. This really is a top notch school full of the best and brightest.

Finally, I would like to thank my parents Joe and Joan who encouraged me to ask questions, and then cheerfully dealt with the consequences.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Many receivers today operate in an interference-limited environment. In a dense environment, the performance of a receiver is limited by interference from multiple signals at the receiver rather than signal-to-noise ratio. In the interference-limited environment there is a diminishing return from investing additional resources into improving traditional receiver parameters such as noise figure. Advanced processing techniques exist that can help recover information that would otherwise be lost using a single-channel receiver [2]. This dissertation investigates some of these techniques.

Interference may originate from many sources: users on the same or adjacent frequency band, unintended emissions, intermodulation, etc. This work investigates interference from cochannel information bearing signals. Cochannel interference is commonly defined as the reception of two or more signals at the receiver overlapping in frequency and in time.

Although multi-user detection is a mature research field [2], little prior research has focused on low-entropy signals. Optimal detection uses all the available information to make a decision. The optimal joint detection techniques are often dismissed due to the complexity of the receiver. Situations exist where this complexity is justified and is the most cost effective way to recover a signal. The field-based maximum *a posteriori* (MAP) joint detection algorithm uses the available *a priori* information to aid in jointly making a decision of what values are sent. Conceptually it is not difficult to understand; the idea is to select the combination of transmitted signals that maximize the *a posteriori* probability of a transmitted field. Let $\boldsymbol{A}$ be a $N \times N_s$ matrix of the transmitted symbol vectors $\boldsymbol{A} = \{\boldsymbol{\alpha_1 \alpha_2} \ldots \boldsymbol{\alpha_{N_s}}\}$, where N_s is the number of signals and N is the length of each vector. The optimum receiver (minimizing the probability of incorrect $\boldsymbol{A}$) is defined as the MAP receiver [3], selecting the most probable $\boldsymbol{A}$ given the received vector $\boldsymbol{r}$. Finding the most likely set of transmitted signals, $\hat{\boldsymbol{A}}$, is now a combinatorics problem; try all 2^{NN_s} permutations of $\boldsymbol{A}$ and select the one that maximizes the *a posteriori* probability. This method is impractical for all but the shortest of messages. This research investigates *efficient* methods of incorporating the information available at a receiver in order to make a decision.

The Automatic Identification System (AIS) is a ship- and land-based tracking and communications system operating in the very high frequency (VHF) maritime band. The primary function of AIS is to provide information for surveillance and the safe navigation of ships [1,4]. The AIS typically sends ship-based tracking messages indicating position and state information at intervals of $2-10$ seconds . This dissertation uses the AIS signal as an example of a signal with the following properties: narrowband, cochannel and low-entropy. The AIS signal is shown to have significant redundancy of information content from message to message. Analysis of the AIS data suggests a 168-bit AIS packet contains approximately 20 bits of new information. This research investigates the low-entropy property of the AIS signal and presents experimental results quantifying this property.

This work develops the Joint Field-Based MAP algorithm in order to achieve the goals stated above. By incorporating *a priori* information from the signal sources, this detector out-performs maximum likelihood (ML) based joint detectors. The field error rate (FER) performance gain is highly dependent on the specific signal characteristics. Here, results are presented for representative signals that show gains of approximately 3 – 18 dB over the current state-of-the-art. This gain does come at a cost. There must be a source of *a priori* information; this may be previous signal receptions or some other source of side information. Without knowledge of one or more of the cochannel signals there will be no gain in performance. There is also a processing cost. Implementing the Joint Field-Based MAP detector is both more computationally costly and more difficult to set up than that of the joint ML detection. The complexity of the Joint Field-Based MAP detector is exponentially dependent on both the number of uses and the field length. A complexity reduction is possible by recognizing that some field values occur with negligible probability. This technique significantly reduces the computational cost while incurring a small penalty in FER performance.

This work focuses on advanced reception techniques that are of particular relevance to the military. Most of these techniques are applicable when there is sufficient extra processing capability available. These techniques are tailored toward asymmetric communications; the scenario where there are existing transmitters and an advanced receiver platform. Although this work uses the AIS signal as an example, it is envisioned that with adequate processing resources these algorithms can be used to improve the bit error rate (BER) performance of many other systems.

This work presents simulation and analytical bounds demonstrating the benefit of using these advanced detection techniques on cochannel signals.

I. INTRODUCTION

As the density of wireless transmitters increases, cochannel interference becomes the major limitation to the performance of a communications receiver. When a receiver is operating in an interference-limited environment additional techniques, beyond improving traditional receiver parameters, must be employed to increase performance. Interference rejection techniques such as adaptive equalization, the constant modulus algorithm, and various non-linear techniques can all be used to help address the cochannel interference problem by attempting to remove the undesired interferer [5]. These interference rejection techniques are not optimal. If multiple receivers are available, array processing techniques can be used to attenuate the undesired signal [6]. Although array processing can provide great performance gains in a cochannel interference environment the requirement for multiple antennas is often impractical. This work focuses on single-antenna joint detection techniques [2]. With joint detection all of the signals are demodulated simultaneously. This method can be shown to satisfy an optimality criterion (minimum probability of error) and does not require multiple antennas.

Some basic definitions should be given here.

The term *cochannel* refers to two or more signals arriving at a receiver overlapping in frequency and time [7]. The signals may all be desired, or may consist of both desired and undesired signals.

The term *narrowband* refers to signals with minimal excess bandwidth i.e., the bandwidth beyond the Nyquist bandwidth of $\frac{1}{2T_s}$, where T_s is the symbol period [8]. It is possible to design a signaling waveform with a large amount of excess bandwidth, many times that of the Nyquist bandwidth, such that a single-channel receiver performs well in the presence of multiple cochannel signals. For example, code division multiple access (CDMA) is a technique that uses waveforms with excess bandwidth to enable simultaneous users. Other multiple access techniques, such as the time division multiple access (TDMA) method, typically do not use a waveform with excess bandwidth.

The term *low-entropy* describes an information source that has other than a uniform distribution (some symbols or sets of symbols are more likely than others) [9]. The low-entropy property of a signal can be exploited to achieve better bit error rate (BER) performance in a detector.

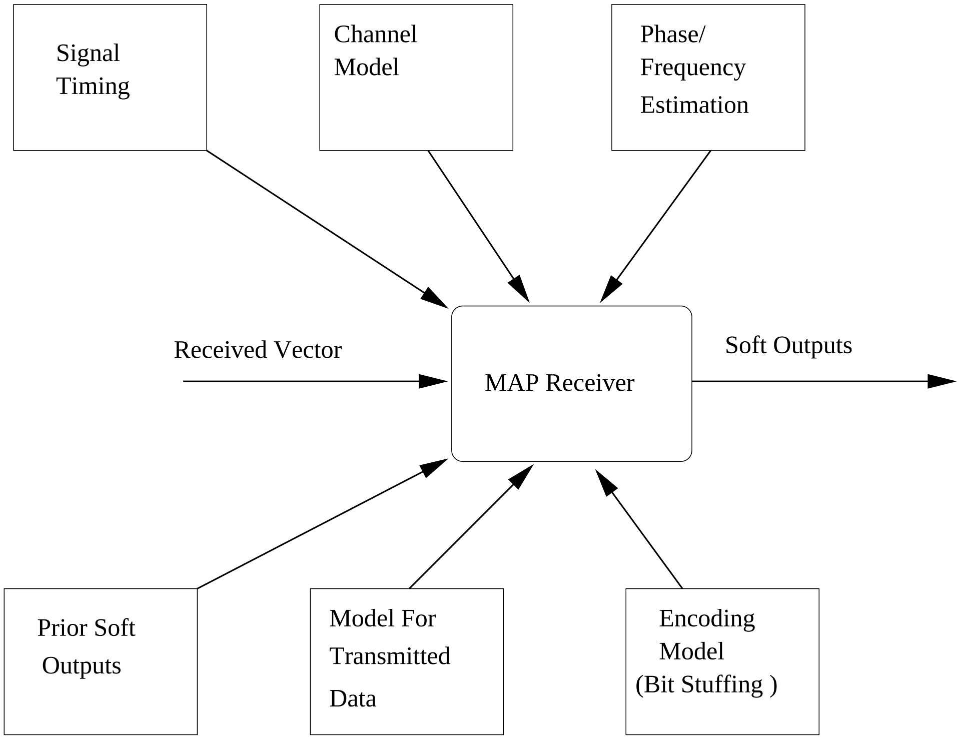


Figure 1.1: Incorporating **All** Knowledge into the MAP Receiver

What is desired is an algorithm that uses the received signal and **all** of the other information available to make a decision. The problem can be formulated as follows: Let $\boldsymbol{\alpha_k}$ be the length N symbol column vector transmitted by the k-th user. Let $\boldsymbol{A}$ be a $N \times N_s$ matrix of the transmitted symbol vectors $\boldsymbol{A} = \{\boldsymbol{\alpha_1}\boldsymbol{\alpha_2}\ldots\boldsymbol{\alpha_{N_s}}\}$, where N_s is the number of signals. The optimum receiver (minimizing the probability of incorrect $\boldsymbol{A}$) is defined as the maximum *a posteriori* (MAP) receiver [3], selecting the most probable $\boldsymbol{A}$ given the received vector $\boldsymbol{r}$.

$$\hat{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}\in\mathcal{A}} \Pr(\boldsymbol{A}|\boldsymbol{r}) \tag{1.1}$$

where $\mathcal{A}$ is the set of possible values for $\boldsymbol{A}$. where $\boldsymbol{r}$ is the received vector (possibly raw samples, possibly pre-processed by some yet to be considered demodulator). What we really want to maximize is

$$\hat{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}\in\mathcal{A}} \Pr(\boldsymbol{A}|\boldsymbol{r};\boldsymbol{e}) \tag{1.2}$$

Where $\boldsymbol{e}$ is the set of other information known about the problem, as illustrated in Figure 1.1.

This dissertation works towards creating an efficient realization of Equation 1.3, ultimately resulting in

$$\hat{\boldsymbol{A}} = \arg \max_{\boldsymbol{A} \in \mathcal{A}_{\updownarrow}} \Pr(\boldsymbol{A}|\boldsymbol{r}; \boldsymbol{e}), \tag{1.3}$$

where $\mathcal{A}_{\updownarrow}$ is a subset of $\mathcal{A}$ containing the values of $\boldsymbol{A}$ with non-negligible probability.

A. NOTATION

Throughout this work the following notation is used unless otherwise stated. Vectors are assumed column vectors and are represented as lower case bold face symbols, $\boldsymbol{a}$. Matrices are written in upper case bold face, $\boldsymbol{A}$. Transpose and complex conjugate transpose are written as $(\cdot)^T$ and $(\cdot)^H$ respectively. Statistical expectation is written as $\mathcal{E}\{\cdot\}$. An estimate of a random variable is written as $\hat{(\cdot)}$. Brackets are used to index into a vector, $x[n]$ is the n^{th} element of the vector $\boldsymbol{x}$; parentheses represent continuous time values, $x(t)$. A vector with a prime, $\boldsymbol{y'}$, indicates that it is an interleaved vector. A probability is represented with $\Pr[\cdot]$; $p(\cdot)$ indicates a probability density function (PDF).

B. COCHANNEL SIGNALS

Cochannel interference is said to occur when two or more signals are received in the same frequency band at the same time. Figure 1.2 illustrates a time-frequency plot with intensity representing the magnitude of two cochannel signals. Notice that the signals in Figure 1.2 overlap in both frequency and time, but portions of the signal do not overlap. This gives hope that the signals can be separated.

Some communications systems such as CDMA use waveforms which are designed to operate when there is cochannel interference [2]. For a system using CDMA, each waveform is separated in code space such that the signals are orthogonal[1], or approximately orthogonal to each other. If the signals are orthogonal then the optimum receiver may be realized by constructing a set of N matched filters, one for each desired signal, followed by N independent decision rules. In this case the optimal decision rule can be derived for each signal by observing the output of a single matched filter.

[1] Orthogonal is used here to mean the inner product of the underling waveforms is equal to 0, i.e., $\int \psi_1 \psi_2 \mathrm{d}t = 0$

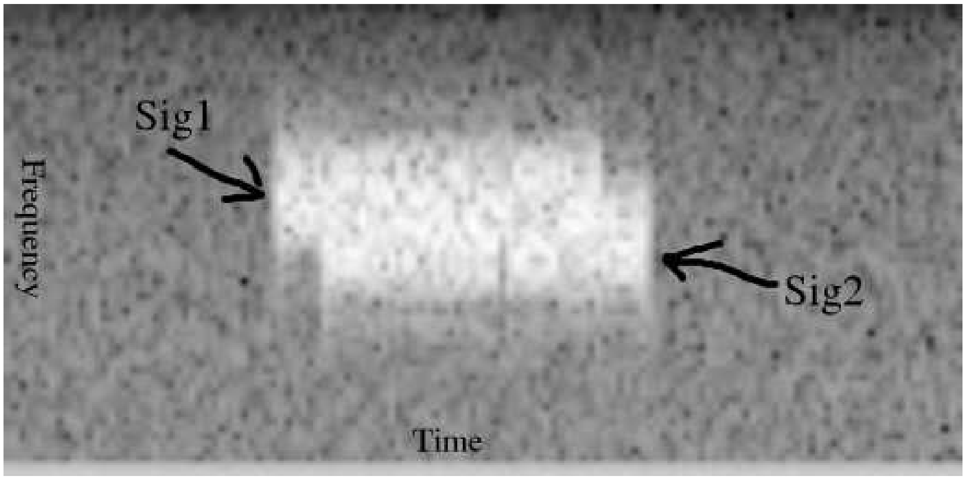


Figure 1.2: Cochannel Signal Example

Other characteristics that allow signal orthogonality (or approximate orthogonality) when the signals occupy the same frequency band at the same time are polarization, phase, and spatial diversity [2, 7, 8]. The focus of this dissertation is the problem of separating cochannel signals when the signals are not directly separable by any of the aforementioned methods.

C. AUTOMATIC IDENTIFICATION SYSTEM (AIS)

An example of a system subject to potential cochannel interference is the reception of the ship-based AIS signal on a high-altitude platform such as an unmanned aerial vehicle (UAV) or satellite [10]. The AIS is a ship/land based tracking/communications system operating in the VHF maritime band. The primary function of AIS is to provide information for the safe navigation of ships and safety of life at sea (SOLAS) [1]. The AIS sends ship-based tracking messages indicating position at update intervals typically between 10 and 2 seconds. The International Maritime Organization (IMO) has ruled that all ships of 300 gross tons and greater must carry an AIS transponder by 2008 [1].

The AIS uses self organized time division multiple access (SOTDMA) to separate users. Each user within line-of-sight (nominally 55 km) attempts to find an unused time slot. Each user effectively creates his own cell centered around his ship. In general, there is no cochannel interference problem for users on the surface of the Earth; users of the same time slot are either not within line-of-sight or are far enough away for the message

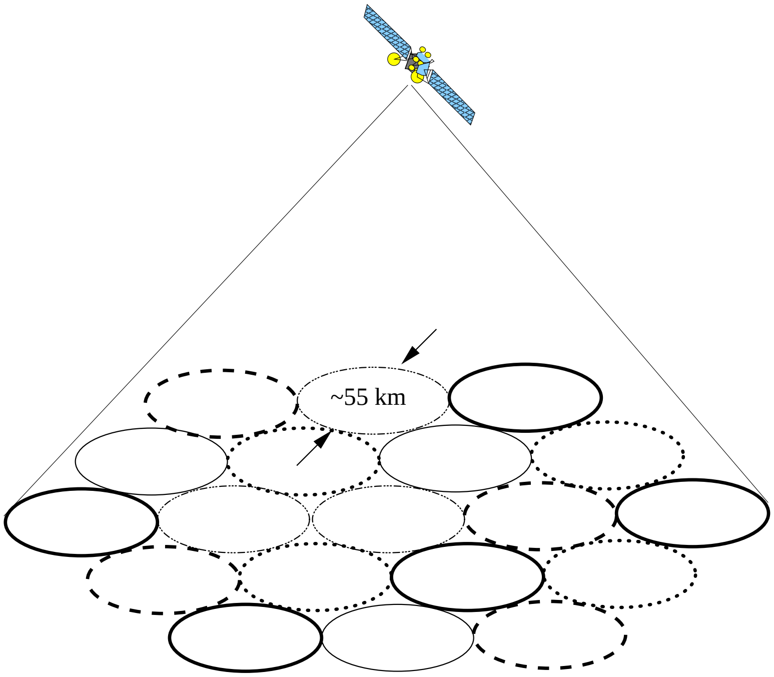


Figure 1.3: Example of Reception from Multiple Cells

importance to be relatively low. A potential interference issue arises when the reception platform has a field-of-view larger than the nominal 55 km as illustrated in Figure 1.3. The coded cells represent areas where the SOTDMA algorithm has separated the signals via TDMA. Cells of the same line style indicate transmissions during the same time slot.

The AIS is not the only signal that can benefit from joint demodulation methods. There are other examples of low-entropy signals that can benefit from these methods such as radio frequency identification (RFID), and signals from sensor networks.

1. Simultaneous Cochannel Signal Receptions

Here, a model is developed in order to investigate the likelihood of cochannel reception. Consider N synchronous users, all within line-of-sight of a single receiver, each with probability p of transmitting at time n. Let the discrete random variable C be the number of cochannel receptions at an instantaneous observation. C is assumed to be a binomial random variable with distribution [11]

$$\Pr(C = k) = p^k (1-p)^{(N-k)} \binom{N}{k}. \tag{1.4}$$

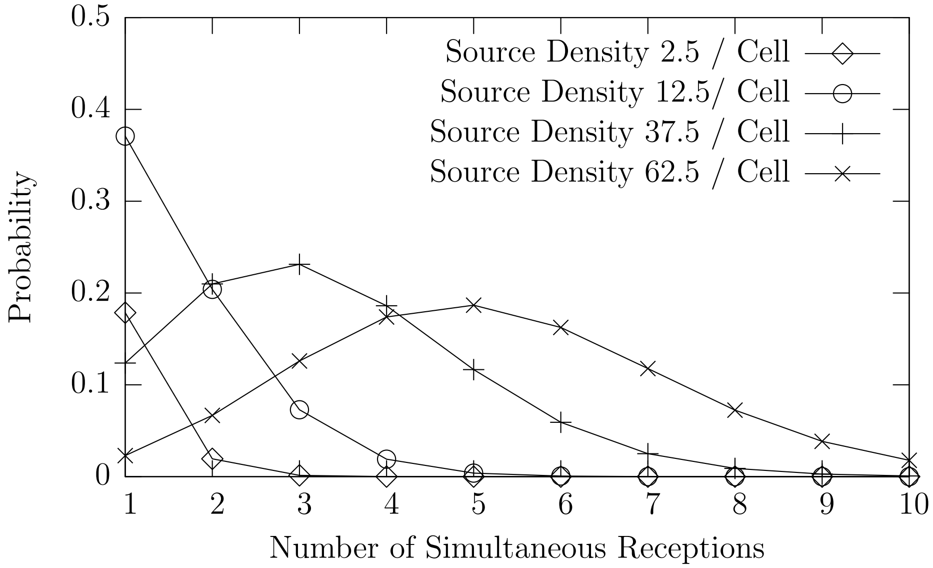


Figure 1.4: Probability of k Simultaneous Receptions

This model describes a shared network of N synchronous users grouped into cells where p is the probability of any user transmitting at time n.

As an example, consider 40 cells, each cell with M users, where each user has a probability of transmission $p = 5/2250 \approx 0.002$. This corresponds to an AIS reporting rate of five reports per minute. Figure 1.4 illustrates the probability of receiving k cochannel transmissions at any one time. The peak of each curve corresponds to the most likely number of receptions for a particular source density. As the number of users per cell increases, the probability of simultaneously receiving more than one signal at a time increases. Notice that for a low density (low number of users per cell) the probability of cochannel reception is low. As the density of users goes up, the most likely number of simultaneous receptions, k, increases. For this example, with 37.5 users per cell, the most likely outcome is simultaneous reception of three signals. The important thing to understand from this example is situations exist where a receiver is likely to receive a small number (less than three or four) cochannel receptions. Although the work in the dissertation generalizes to N users the focus is on two or three users.

D. COCHANNEL SIGNAL RECOVERY TECHNIQUES

In order to successfully jointly demodulate cochannel signals there must be some distinguishing feature between them. In each case the distinguishing feature leads to underlying signaling waveforms which have a normalized cross-correlation coefficient [2] with magnitude less than one.

1. Frequency Separation From Doppler

If the transmitters of the signals $s_i(t)$ have different relative velocities with reference to the collection platform they will appear shifted in frequency with respect to each other at the receiver. This frequency separation can be exploited in the receiver. For narrowband, high frequency signals with a large difference in relative velocities, it is quite possible that the relative motion would completely separate the signals at the receiver such that they are orthogonal. As an example, the space shuttle at low earth orbit (LEO) receiving the AIS signal would see a maximum frequency separation of approximately 4 kHz. While this is enough to assist in joint demodulation, it is not enough to separate the signals such that they are orthogonal, so single-channel demodulation techniques are not effective.

2. Multiple Receivers

Multiple receivers provide an additional level of diversity that may be exploited. Much work has been published in the area of beam-forming with multiple coherent receivers; see [6, 8] for a thorough overview. Although this is a promising area, multiple receivers result in additional cost are sometimes impractical, therefore this dissertation will focus on single-antenna architectures. Single-antenna receiver architectures for cochannel interference mitigation are often referred to as Single-Antenna Interference Cancellation (SAIC) [7].

3. *A priori* Symbol Probabilities

Another promising area is taking advantage of *a priori* symbol probabilities when performing the demodulation. If data contained in the signal $s_i(t)$ are not random but known or partially known, a joint receiver structure can be designed that incorporates this knowledge.

Table 1.1: Binary Representation of Position

Binary	**Position (m)**
00000000	±0 meters
00000001	+1 meters
00000010	+2 meters
11111111	−1 meters
11111110	−2 meters
...	...

Consider a transmitted position value (x). The position value, in meters, is mapped to an 8 bit two's complement representation (Table 1.1). The bits are mapped to binary symbols $\alpha \in \{-1, 1\}$, and transmitted through an additive white Gaussian noise (AWGN) channel. This position value is transmitted via a binary antipodal modulation scheme. The transmitted signals are then received along with AWGN, and are coherently detected.

First, suppose there is no *a priori* information available for the position. The probability of receiving an incorrect position P_e is given by [8]

$$P_e = 1 - (1 - P_{\text{bit error}})^8, \tag{1.5}$$

where $P_{\text{bit error}}$ is given by [8]

$$P_{\text{bit error}} = Q\left(\sqrt{\frac{E_d}{2N_0}}\right), \tag{1.6}$$

$Q(x)$ is defined as

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2}\,\mathrm{d}\tau, \tag{1.7}$$

and E_d is the squared Euclidean distance between transmitted symbols for binary 1 and binary 0

$$E_d = (\alpha_0 - \alpha_1)^T(\alpha_0 - \alpha_1) \tag{1.8}$$

where α_0 and α_1 are the symbols used to represent a binary 0 and a binary 1 respectively, and N_0 is the one-sided noise power spectral density.

Now suppose it is known that the actual position value is either 15 or 16 meters, $x \in \{15, 16\}$, with equal likelihood. The binary representation of the message therefore is either

$$00001111$$

or

$$00010000$$

Although these two values differ in information by one bit, the Hamming distance[2] in the "code words" is 5 bits. Not only has the problem been reduced from selecting one of 2^8 sequences to that of selecting one of 2^1 sequences, the sequences are further separated in Euclidean space, making the decision less error prone. Here, the probability of position error is only

$$P_e = Q\left(\sqrt{\frac{5E_d}{2N_0}}\right), \tag{1.9}$$

where E_d is defined as the squared Euclidean distance between the symbols representing a binary 1 and a binary 0 as in (1.8). This is a significant improvement in probability of error over the first case.

Now suppose the position was known to be either 16 or 17 meters, $x \in \{16, 17\}$, with equal likelihood. The positions are represented as

$$00010000$$

and

$$00010001$$

respectively. In this case, the Hamming distance between words is only one yet, by using this *a priori* knowledge of the transmitted values the decision is selecting one of 2^1, instead of selecting one of 2^8 possibilities. The probability of position error is then given by

$$P_e = Q\left(\sqrt{\frac{E_d}{2N_0}}\right), \tag{1.10}$$

where E_d is given by (1.8).

[2] Given two strings of equal length, the Hamming distance is the number of positions that the strings differ [12]

There are two challenges to exploiting *a priori* probabilities in a receiver. The first is the calculation of the *a priori* probabilities, and then mapping this information to the received symbols. Estimates for physical quantities such as position, velocity, time, etc. can be performed with a variety of predictors. The second is the increased complexity of the receiver. For the example above where the transmitted value is one of two equally likely position values, there is little increase in complexity. For a large number of possible position values that are not equally likely, the decision rules become complex.

4. Time-of-Arrival Offset

Many communications systems such as AIS are packet-oriented. For some collection scenarios even packets in a synchronized network could arrive with a time offset. Overlapping packets should be treated as a special case. Three situations are illustrated in Figure 1.5: clean reception of a packet, clean reception of the beginning or end of a packet, and overlap of two packets.

Clean reception can be used for calculation of symbol probabilities. The clean beginning or end of a packet can be used for phase and frequency estimation. The region of overlap is where the joint detection techniques are used to recover the signals.

5. Multiple Message Integration

If a data source transmits the same information in multiple messages, soft decisions from each message can be combined to obtain a more reliable estimate of what was transmitted.

E. SIMPLE MITIGATION EXAMPLE

It is instructive introduce a simple example of cochannel interference mitigation. Consider two synchronous symbol streams:

$$\begin{array}{lcrrrrr}
n & = & 0 & 1 & 2 & 3 & 4 \\
\boldsymbol{\alpha}^{(1)}[n] & = & \{-1, & +1, & +1, & -1, & -1\} \\
\boldsymbol{\alpha}^{(2)}[n] & = & \{-1, & -1, & +1, & +1, & -1\} \\
\boldsymbol{r}[n] = \boldsymbol{\alpha}^{\mathbf{(1)}} + \boldsymbol{\alpha}^{\mathbf{(2)}} & = & \{-2, & 0, & +2, & 0, & -2\}
\end{array}$$

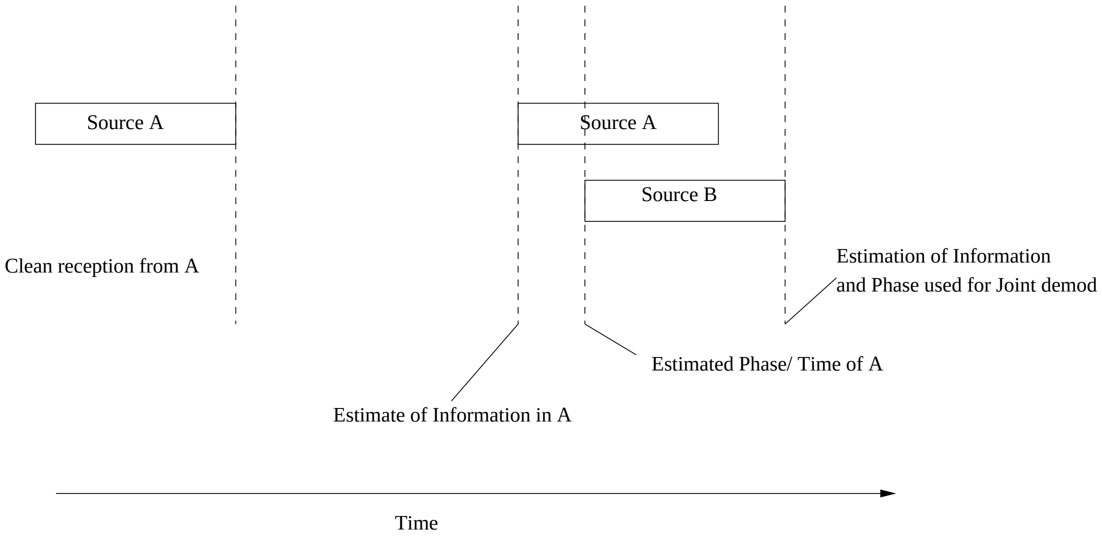

Source A
Source A
Source B
Clean reception from A
Estimation of Information
and Phase used for Joint demod
Estimated Phase/ Time of A
Estimate of Information in A
Time

Figure 1.5: Packet Overlap Example

Given the received stream $\boldsymbol{r}$ a reasonable estimate of $\boldsymbol{\alpha}^{(\mathbf{1})}$ and $\boldsymbol{\alpha}^{(\mathbf{2})}$ is

$$\hat{\boldsymbol{\alpha}^{(\mathbf{1})}} = \hat{\boldsymbol{\alpha}^{(\mathbf{2})}} = \{-1, \pm 1, +1, \pm 1, -1\}. \tag{1.11}$$

There are two locations for possible errors in each of the received streams, the locations at $n = 0$ and $n = 3$. At both of these locations, if the bit for one stream ($\boldsymbol{\alpha}^{(1)}$ or $\boldsymbol{\alpha}^{(2)}$) is known the bit for the other could be determined.

Now suppose that the data sequences $\boldsymbol{\alpha}^{(\mathbf{1})}$ and $\boldsymbol{\alpha}^{(\mathbf{2})}$ are sampled at twice the symbol rate such that:

$$\begin{array}{rcrrrrrrrrrr} n & = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \boldsymbol{\alpha'}^{(1)}[n] = \alpha^{(1)}\left[\lfloor \frac{n}{2} \rfloor\right] & = & \{-1, & -1, & 1, & 1, & 1, & 1, & -1, & -1, & -1, & -1\} \\ \boldsymbol{\alpha'}^{(2)}[n] = \alpha^{(2)}\left[\lfloor \frac{n}{2} \rfloor\right] & = & \{-1, & -1, & -1, & -1, & 1, & 1, & 1, & 1, & -1, & -1\} \\ \boldsymbol{r'} = \boldsymbol{\alpha'}^{(\mathbf{1})} + \boldsymbol{\alpha'}^{(\mathbf{2})} & = & \{-2, & -2, & 0, & 0, & 2, & 2, & 0, & 0, & -2, & -2\} \end{array}.$$

This provides no more information than the first case.

Now suppose that the symbol timing from s_1 and s_2 is received such that s_2 is delayed by $T_s/2$ and this delay is known at the receiver. This delay could be caused by unsynchronized transmitters or path length differences in synchronized transmitters.

$$\begin{array}{rcrrrrrrrrrrr}
n & = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\alpha'^{(1)}[n] = \alpha^{(1)}\left[\lfloor \frac{n}{2} \rfloor\right] & = & \{-1, & -1, & 1, & 1, & 1, & 1, & -1, & -1, & -1, & -1 & 0\} \\
\alpha''^{(2)}[n] = \alpha^{(2)}\left[\lfloor \frac{n-1}{2} \rfloor\right] & = & \{0 & -1, & -1, & -1, & -1, & 1, & 1, & 1, & 1, & -1, & -1\} \\
\boldsymbol{r''} = \boldsymbol{\alpha'^{(1)}} + \boldsymbol{\alpha''^{(2)}} & = & \{-1, & -2, & 0, & 0, & 0, & 2, & 0, & 0, & 0, & -2, & -1\}
\end{array} .$$

The two transmitted sequences may now be resolved from the received sequence. An algorithm for recovering $\boldsymbol{\alpha^{(1)}}$ and $\boldsymbol{\alpha^{(2)}}$ is as follows.

initialization:

$$\hat{\alpha}^{(1)}[0] = r''[0] \tag{1.12}$$

$$\hat{\alpha}^{(2)}[0] = r''[1] - \hat{\alpha}^{(1)}[0] \tag{1.13}$$

recursion:

$$\hat{\alpha}^{(1)}[k] = r''[2k] - \hat{\alpha}^{(2)}[k-1] \tag{1.14}$$

$$\hat{\alpha}^{(2)}[k] = r''[2k+1] - \hat{\alpha}^{(1)}[k] \tag{1.15}$$

This example can be extended to continuous time digital communications signals received by a bank of matched filters sampled at the symbol rate, followed by a joint detection algorithm. Because of the recursion, a single error at step n can lead to multiple errors for later steps. The propagation of errors is investigated further in Chapter V and Chapter VII.

F. IMPORTANCE OF THIS RESEARCH

Multi-user detection is a mature research field, yet there are still some areas where new contributions can be made; little work has focused on low-entropy signals. This work focuses on the reception of low-entropy signals from simple (no advanced coding techniques) transmitters with an advanced single-antenna receiver. The first assumption is that the transmitter is not generating signals intended for joint reception, and the transmitted signals have some degree of predictability. The second assumption is that the receiver has processing resources available to implement advanced algorithms. This research extends prior techniques of joint reception to this scenario, allowing for successful detection of signals with a lower error rate than was previously possible.

G. CONTRIBUTIONS

The major original contributions of this work are the following:

- A maximum likelihood sequence detection (MLSD) for a bit stuffed data source. This detector gives superior BER performance compared to the ML symbol-by-symbol detector for the same signal-to-noise ratio (SNR) (a gain of 0.2 – 1.0 dB).
- An AIS communications state predictor and demonstration of the performance for a representative data set. The communications state field in the AIS message contains important synchronization information that can be used to predict when it is likely another similar message will be sent.
- Prediction method for AIS field data; including the latitude and longitude fields, and simulation of the performance of the prediction methods. The 27 bit longitude field is shown to contain approximately four bits of new information once a single message has been received.
- A Joint Field-Based MAP detector for cochannel signals. This detector results when *a priori* information is available for fields of data.
- A method of state reduction; by ignoring states with negligible probability (similar to the algorithms presented in [13, 14]). State reduction becomes necessary for the general solution to make the problem computationally feasible.
- An analysis of time-of-arrival variation for AIS packets. Accurate estimation of timing parameters is an important component of the optimum detector.
- An analysis of the reporting accuracies for the state information fields in an AIS message, for example, the reporting precision for the longitude and latitude fields are found to vary from AIS to AIS.

H. DISSERTATION ORGANIZATION

The remainder of this dissertation is organized as follows: Chapter II provides an introduction to some fundamental concepts of joint detection and a full literature review. Chapter III provides a complete overview of the AIS and presents some experimental results demonstrating the “low-entropy” property of this signal. Chapter IV provides an overview of prediction methods and develops predictors for the AIS signal specifically. It provides experimental results showing the performance of the predictors on collected data. Chapter V develops the MLSD for a bit-stuffed data source; bounds on BER performance are developed and compared to simulation. Chapter VI is an overview of the multicategory classification problem that results from a field-based MAP detector based on field *a priori* information. The single-channel detector is then extended to the special case of synchronous cochannel reception. Bounds on performance for the single-channel and synchronous cochannel example are developed and compared with simulation. Chapter VII extends the results of Chapter VI to the general case of asynchronous signals. In this extension, a time-varying trellis is developed to carry forward *a priori* information from each field. An efficient trellis traversal method is presented. Finally, Chapter VIII summarizes the results of this work, provides guidance for the practical implementation of some of this ideas from this work, and discusses areas of future research.

II. PRIOR WORK

This chapter provides a history of prior work in the area of cochannel interference mitigation and multi-user detection. A review of other authors' interest in AIS is presented. An overview of prior research in both coherent and non-coherent Gaussian minimum shift keying (GMSK) receivers is developed. Finally, a review of the fundamentals of communications theory required later in this work is provided.

A. AIS

In this dissertation, AIS is the exemplar low-entropy signal considered for joint detection. There has been some interest in collection of AIS from high-altitude unmanned aerial vehicle (UAV) and from space [15–17]. In [17], the authors discuss the future of maritime surveillance using micro-satellites; they investigate collecting radar pulses and/or AIS signals along with satellite-based synthetic aperture radar (SAR) imagery. The authors also discuss the Norwegian micro-satellite NSAT-1 which is aimed at demonstrating detection and direction finding on maritime X-band radars.

As more ships are equipped with AIS and as the next class of low power AIS (Class B systems) become available, cochannel interference is expected to become more of a problem [18]. In [17], the authors discuss the potential interference issues associated with collection of AIS from space. A Norwegian student project, NCUBE, is looking into the feasibility of collecting AIS from space. In [17], the authors hint at "advanced on-board processing" to enable a satellite to handle more messages than would otherwise be possible from a single receiver.

In [16], the authors discuss some modifications to AIS to support Long Range Identification and Tracking (LRIT) in an effort to increase global maritime domain awareness.

B. GMSK

GMSK is the modulation scheme used in the AIS. GMSK is a popular modulation technique when spectral efficiency and constant modulus is desired. A constant modulus signal requires a less complicated amplifier stage at the transmitter making this an attractive scheme. Murota [19], investigates GMSK modulation for what has become today's Global System for Mobile Communications (GSM) cell phone systems. Murota examines spectrum manipulation of minimum-shift keying (MSK) by introducing a pre-modulation Gaussian low-pass filter (LPF). Murota explores the general properties of GMSK, error performance, and bandwidth consumption, and investigates some implementations.

GMSK is a continuous-phase modulation (CPM) signal with a Gaussian filter, $g(t)$. The complex baseband representation of a CPM signal is given by:

$$z(t) = A \exp(j\phi(t; \boldsymbol{\alpha})), \tag{2.1}$$

where A is the signal amplitude, assumed to be one hereafter, and $\phi(t; \boldsymbol{\alpha})$ is the phase term whose current value depends on all previous symbols (α_n),

$$\phi(t; \boldsymbol{\alpha}) = h\pi \sum_n \alpha_n q(t - nT_s). \tag{2.2}$$

Here, T_s is the symbol period and h is the modulation index, assumed to be a rational number. The modulation index h is proportional to the maximum phase change per symbol period. The maximum phase change per symbol period is $h\pi/2$ radians. The information bits are mapped to symbols α_n

$$\alpha_n \in \{-1, +1\}, \tag{2.3}$$

and $q(t)$ is the integrated pulse shape

$$q(t) = \int_0^t g(\tau)\,\mathrm{d}\tau. \tag{2.4}$$

For GMSK $g(t)$ is the pulse response of a Gaussian filter,

$$g(t) = \mathrm{rect}(t/T_s) * \frac{1}{\sqrt{2\pi\sigma^2 T_s^2}} \exp\left(\frac{-t^2}{2\sigma^2 T_s^2}\right), \tag{2.5}$$

where $\sigma = \sqrt{\ln(2)}/2\pi BT_s$ [20], $*$ denotes convolution, and $\mathrm{rect}(t/T_s)$ is defined as

$$\mathrm{rect}(t/T_s) = \begin{cases} 1/T_s & |t| < T_s/2 \\ 0 & |t| > T_s/2 \end{cases}. \tag{2.6}$$

The pulse response $g(t)$ can be written as a difference of Q functions[1],

$$g(t) = \frac{1}{2T_s}\left[Q\left(2\pi BT_s\frac{t-T_s/2}{T_s\sqrt{\ln(2)}}\right) - Q\left(2\pi BT_s\frac{t+T_s/2}{T_s\sqrt{\ln(2)}}\right)\right], \tag{2.8}$$

where BT_s is the time-bandwidth product of the pulse response $g(t)$, and B is the 3 dB bandwidth of the Gaussian filter.

Typically $g(t)$ will be truncated to L symbol periods, depending on the value of BT_s, such that $g(t)$ is nonzero on the interval $[0, LT_s]$ and symmetric around $LT_s/2$. The integrated phase function, $q(t)$, is nonzero on the interval $[0, \infty)$ and is constant $1/2$ for values of $t > LT_s$ [8].

CPM signals are in general non-linear; they can not, in general, be represented as a superposition of amplitude modulated pulses. Laurent [21] shows that for finite pulse response, $g(t)$, any binary CPM signal can be represented as a superposition of 2^{L-1} amplitude modulated pulses. This is an extension of the well known representation of MSK as offset quadrature phase shift keying (OQPSK). In [22], the authors develop a minimum mean-square error (MMSE) pulse representation of CPM signals, and provide a supplementary proof of the Laurent representation. In [23], the authors investigate modulator and demodulator simplifications based on the Laurent representation when the modulation index $h = 1/2$.

In [24], the authors show that for small modulation index, approximately less than 0.5, the performance gain from using additional pulses after the first pulse in the Laurent representation is minimal. In [25], the authors compare the performance of a MLSD using only the “main pulse”, referred to as the AMP-MLSD. They show the BER performance degradation from using only the main pulse in the MLSD is a fraction of a dB. The importance of the Laurent representation is that only the first few terms in the sum are

[1] The Q-function is defined as

$$Q(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2}\, \mathrm{d}\tau \tag{2.7}$$

needed, thus reducing the complexity of a receiver. A full description of the Laurent representation is provided in Appendix A.

C. SYNCHRONIZATION

Carrier phase and frequency information are required for coherent detection of GMSK signals. Typically this is accomplished with an initial estimate from a known training sequence followed by some form of phase tracking algorithm.

Boashash [26], provides an exhaustive overview of estimation techniques for instantaneous frequency and, by extension, phase. He covers various maximum likelihood estimate (MLE) techniques as given in Kay [27], and model-based methods such as modeling the phase term as a linear equation and using least squares to develop an estimate of frequency and phase as in [28]. In [28], Tretter shows that for high SNR the frequency estimate based on fitting a line to the unwrapped phase term has the same variance as the Cramer-Rao bound. In [29], Hansen provides an excellent tutorial on phase and frequency estimation and provides additional examples.

In [30], the authors investigate a method of jointly estimating both data and timing in the presence of intersymbol interference (ISI). The authors model the ISI as a Markov chain, and model the timing variation as a random walk.

Rice, et al. [31], discuss data-aided phase estimation for GMSK using only the first Laurent pulse. They show that the bias of the ML estimate in [32] is due to the edge effects of the estimation in that paper. Mengali and D'Andrea [32] serves as an excellent reference for synchronization techniques of sampled signals. It provides a thorough theoretical background without losing sight of implementation details. The authors cover both optimal and suboptimal techniques, and provide closed form solutions for performance where practical.

Harris, et al. [33], discusses timing synchronization using polyphase filters. The authors address the problem of re-sampling and interpolating after a matched filter by implementing the matched filter as a polyphase filter bank.

In [34], the authors discuss joint carrier-phase and symbol-timing for coherent CPM receivers.

In [35], the authors show that the Cramer-Rao bound (CRB) for data aided timing or carrier phase recovery varies with training sequence.

There are various methods that combine symbol and channel estimation into the same problem. Reader, et al. [36] introduce a Blind Maximum Likelihood Sequence Detection (BMLSD) where the estimation of the channel parameters is included in the derivation of the MLSD. In [37], the authors investigate the blind acquisition characteristics of Per-Survivor Processing (PSP). They show that short-term poor performance of PSP is often from transmitted sequences that do not allow the channel to be determined.

D. GMSK RECEIVERS

GMSK signals may be demodulated coherently or non-coherently [38]. Huang, et al. [39], provide an excellent overview of coherent and non-coherent receivers for GMSK. They propose a fast Fourier transform (FFT) based architecture for phase, carrier, and symbol timing recovery along with demodulation both coherently and non-coherently. They perform simulations for receiver performance for various estimation errors in phase, frequency, and symbol timing. In [40], the authors analyze decision feedback for one-bit differential GMSK detectors. In [41], the authors analyze the performance of one-bit differential detection of GMSK signals. In [42], the authors analyze the performance of two-bit differential detection. In [43], the authors develop a simplified non-coherent detector that uses a simplified phase pulse.

After Laurent [21] proposed representing a CPM signal as a linear combination of amplitude modulated pulses, many authors investigated using this method to develop receivers based on all or a fraction of the Laurent pulses. In [44], Kaleh uses the Laurent representation of a CPM signal to develop the optimal receiver as a set of N complex matched filters followed by a Viterbi detector. He then demonstrates that by using a subset of the N complex matched filters a simplified Viterbi receiver can be developed that approaches the performance of the optimum receiver. He then provides an example using GMSK with $BT = 0.25$. In [45], the authors develop a four-state maximum-likelihood sequence estimation (MLSE) GMSK demodulator. They achieve state reduction by only using the first Laurent pulse and by using a whitening filter prior to the MLSE.

Many authors have looked at differential detection of GMSK using decision feedback [46, 47]. There are many proposed non-coherent detection methods for GMSK signals. In [48], the authors propose a non-coherent differential detection method based on maximum-likelihood sequence estimation of the transmitted phase. In [49], the authors investigate the performance of decision feedback equalization and non-linear data-directed estimation as alternatives to MLSE in the presence of adjacent channel interference.

In [47], the authors investigate GMSK differential detectors with decision feedback for cochannel interference (CCI) channels. The authors show the superior performance of a 2-bit differential detector compared to a 1-bit or 2-bit detector with one interferer and a carrier-to-interferer power ratio of 11 dB.

E. MLSE

MLSE is commonly used as a detection method for signals where ISI is introduced either at the transmitter or in the channel. The Viterbi Algorithm is an efficient method for finding the shortest (or the longest) path through a layered graph or trellis. The algorithm has been put to use as an effective method to compute the MLSE.

In [50], Sklar provides an excellent tutorial on using the Viterbi decoding algorithm for channel equalization. He introduces the Viterbi decoding algorithm, discusses computational complexity and applies it to channel equalization.

The Forney MLSE is introduced in [51]. This receiver structure is a matched filter followed by a whitening filter to remove the statistical correlation of the noise from sample to sample. The output of the whitening filter is then applied to a sequence estimator such as the Viterbi decoder with branch metrics of Euclidean distance. The matched filter, sampler and discrete-time whitening filter are often together referred to as the whitening matched filter [8, 52]. In [53], a new upper bound is developed that is shown to be tighter than the Forney bound developed in [51].

The Ungerboeck receiver, introduced in [54], does not require a whitening filter. It consists of a matched filter and sampler, followed by a sequence estimator such as the Viterbi decoder with a modified branch metric. The Ungerboeck receiver is developed from the same ML principle as the Forney receiver and will have identical performance in the same channel. The Ungerboeck receiver uses the symmetry of the autocorrelation function of the channel in development of the branch metric. This modified branch metric negates the requirement of the whitening filter in the development of the MLSD.

A unification of both the Forney and Ungerboeck receivers is presented in [55]. In this work, each receiver is derived from the other providing further insight into each algorithm. The authors show that the whitening operation in the Forney receiver is canceled by the Euclidean distance metric, leaving the modified metric of the Ungerboeck receiver.

F. COCHANNEL INTERFERENCE

There are many terms used in the literature that generically refer to cochannel interference. The term multiple access interference (MAI) is commonly used to describe interference from users within the same network or cell. CCI is often used to describe interference from a different network or different cell. In this dissertation cochannel interference is used to describe both phenomena.

Shnidman [56], was the first to point out that cochannel interference and ISI are essentially an identical phenomena. Both CCI and ISI lead to multiple symbol contributions at the output of a filter matched to a single symbol period. Kaye [57], further investigated the problem of multiple signal detection over multiple channels.

Single-antenna cochannel interference mitigation techniques can roughly be placed into two different categories. The first category consists of filter based approaches including subspace methods that involve filtering out the interference. These non-linear techniques rely on the signal and interferer being somehow linearly separable. The second category consists of multi-user detection techniques such as joint maximum likelihood sequence estimation (JMLSE) and successive interference cancellation. These techniques jointly estimate all of the signals; separation is achieved through the joint estimate. The authors in [7, 58, 59] provide a good overview of these techniques.

1. CCI Filtering Methods

There has been much work in the area of reducing the complexity of the optimal joint detector. The computational complexity of the optimal detector grows exponentially in both ISI length and number of signals [2], making it practical only for a small number of simultaneous users.

In [60–62], the authors develop an interference cancelling matched filter by using known training sequences. In [63], the authors investigate the cancellation of unwanted interfering signals based on second-order statistics. The constant-modulus algorithm and the phased-locked loop are extended for use in blind multi-user detection in [64]. In [65], the authors provide a performance comparison of coherent and limiter-discriminator GMSK detectors in the presence of a single interferer for various carrier-to-interferer ratios and various bandwidth-time products.

Pukkila, et al. [66, 67], investigate SAIC for GSM where the authors assume the interference is constant modulus, and they assume they do not have channel state information. This technique is known as the Constant Modulus - Single-Antenna Interference Cancellation (CM-SAIC) algorithm.

2. Joint Detection

Optimum joint detection techniques are often rejected because the complexity of the computation is exponential in the number of users. For detection of only two or three users the optimal techniques become viable. For narrowband signals, SNR limitations would typically limit the joint detection of multiple users to no more than two or three users.

In [68–70], the JMLSE is developed for continuous-phase modulated signals. This involves representing the CPM signals using the Laurent representation [21], and following the methods developed for linear modulation methods.

Polydoros and Chugg [71] discuss joint reception of narrowband signals based on work in [2]. The authors investigate iterative-detection techniques and spend some effort discussing complexity-performance tradeoffs.

In [72], the authors develop a lower bound for bit error probability for the JMLSE with perfect channel knowledge.

Van Etten [59] describes a method of extending the Viterbi algorithm for ISI to handle multiple cochannel signals. He termed this the “Vector Viterbi Algorithm”.

Verdú [73] uses the Ungerboeck formulation of the MLSE to form an optimal k-user multi-user detector for asynchronous signals.

In [74], the authors describe a blind adaptive joint detector. The detector jointly estimates both channel and transmitted data for each signal. The authors propose a $T_s/2$ sampler. Their detector is based on [75].

Giridhar, et al., [76] describe an optimal method for jointly demodulating cochannel signals termed the joint maximum *a posteriori* sequence detector (JMAPSD). They develop an optimal joint demodulator and investigate the performance of suboptimal receivers.

In [77], the authors develop the optimum joint MLSE for GSM signals. To enable coherent detection, the authors develop a method to jointly estimate channel parameters based on a known training sequence.

In [78], the authors develop a model-based method for CPM demodulation in the presence of cochannel interference. The authors use modern spectral estimation techniques on a symbol-by-symbol basis in order to estimate instantaneous frequency. They provide an excellent summary of standard CPM demodulation techniques.

3. Suboptimal Joint Detection

In [79], a receiver structure consisting of a bank of matched filters followed by a whitening filter and a vector Viterbi algorithm is proposed. In [80], the authors propose an orthogonal multi-user receiver. The proposed receiver does not require knowledge of signal amplitude or noise power. Ranta, et al., [81] propose a method of exploiting the independently fading multipath channels to provide separation of transmit waveforms to be used in joint demodulation.

4. Development of the JMLSE

Below is an introduction to JMLSE. It is well known that the output of a single matched filter generates a sufficient statistic for detection of a signal in Gaussian noise [8]. Consider the detection of multiple synchronous signals in noise with a single receiver. The received signal $r(t)$ is the summation of these multiple signals plus noise:

$$r(t) = \sum_{k=1}^{N_s} s_k(t) + n(t). \tag{2.9}$$

A simple strategy for detecting the multiple signals is to form a bank of matched filters each followed by a decision threshold. If the signals are orthogonal and the symbols equally likely, this structure could be shown to minimize the probability of error. If the signals are not orthogonal, this method will not minimize probability of error. For signals that are highly correlated or slightly correlated with large differences in received power, this method will perform very poorly. The outputs $(y_1, y_2, \ldots, y_{N_s})$ of N_s matched filters matched to each of the signals are sufficient statistics for estimating the symbols $(\alpha_1, \alpha_2, \ldots, \alpha_{N_s})$, but y_1 is not a sufficient statistic for estimating α_1 [2]. This is because, in general, the signals s_k, are not mutually orthogonal, so in general the decision statistics y_k are not independent, even when $n(t)$ is AWGN. It is for this reason that joint detection is required. W. van

Etten [82] demonstrates that the joint detection problem is an extension of the ISI problem. He was the first to show that the outputs of multiple matched filters matched to each of the cochannel signals form sufficient statistics for joint estimation of the transmitted signals.

For asynchronous signals, the matched filter outputs at a particular sample time are not sufficient statistics for determining the symbols transmitted at that same instance; the matched filter outputs from the entire duration of the signal $(\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_{N_s}})$ must be used in the estimation of the symbol vectors $(\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \ldots, \boldsymbol{\alpha_{N_s}})$ [2].

5. General N_s Signal Case

Below we the joint MLSD is developed for the N_s-user case using a complex baseband representation and a modified metric introduced by Ungerboeck in [54] for an ISI channel. The development closely follows the work by Verdú [2, 53, 73], Murphy [68] and examples in [8].

Consider the reception of N_s signals in AWGN

$$r(t) = \sum_{k=1}^{N_s} s_k(t) + z(t). \tag{2.10}$$

Where $z(t)$ is AWGN and $s_k(t)$ is represented as

$$s_k(t) = \sum_i A_k \alpha_k[i] \psi_k(t - iT_s - \tau_k), \tag{2.11}$$

where the unit-energy signal pulse, $\psi_k(t)$, is only nonzero on the interval $[0, T_s]$, τ_k is the time offset, and A_k is the amplitude for the k^{th} signal.

Form a set of matched filters whose outputs at the sample time are

$$y_k[i] = \int_{-\infty}^{\infty} r(t) \psi_k(t - iT_s - \tau_k) \, \mathrm{d}t, \tag{2.12}$$

where ψ_k are the unit-energy signal pulses from (2.11). The matched filter outputs, $(\boldsymbol{y_0}, \boldsymbol{y_1}, \ldots, \boldsymbol{y_{N_s-1}})$, are a linear combination of all the signals' individual responses to the filter: the desired response plus the response of $2 \times (N_s - 1)$ other symbols from the other channels. For pulses with a duration of one symbol period, the matched filter outputs will consist of contributions from the symbol just before and the symbol just after from each other channel. This is shown in Figure 2.1. The output of the matched filters can be written as

$$\begin{aligned} y_k[i] &= A_k\alpha_k[i] \\ &+ \sum_{j<k} A_j\alpha_j[i] \int_0^{T_s} \psi_k(t)\psi_j(t-(\tau_j-\tau_k))\,\mathrm{d}t \\ &+ \sum_{j<k} A_j\alpha_j[i+1] \int_0^{T_s} \psi_k(t)\psi_j(t-T_s-(\tau_j-\tau_k))\,\mathrm{d}t \\ &+ \sum_{j>k} A_j\alpha_j[i-1] \int_0^{T_s} \psi_k(t)\psi_j(t+T_s-(\tau_j-\tau_k))\,\mathrm{d}t \\ &+ \sum_{j>k} A_j\alpha_j[i] \int_0^{T_s} \psi_k(t)\psi_j(t-(\tau_j-\tau_k))\,\mathrm{d}t \\ &+ z_k[i], \end{aligned} \tag{2.13}$$

where $z_k[i]$ is a complex Gaussian random variable with variance $\sigma^2 = N_0$. The Gaussian random process formed by interleaving the noise components from the output of each matched filter is given by $\boldsymbol{z}[i] = \begin{bmatrix} z_1[i] & z_2[i] & \ldots & z_{N_s}[i] \end{bmatrix}^T$ and has autocorrelation matrices

$$\mathcal{E}\{\boldsymbol{z}[i]\boldsymbol{z}^T[j]\} = \begin{cases} \sigma^2\boldsymbol{R}^T[1], & \text{if} j = i+1 \\ \sigma^2\boldsymbol{R}^T[0], & \text{if} j = i \\ \sigma^2\boldsymbol{R}[1], & \text{if} j = i-1 \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{2.14}$$

where $\boldsymbol{R}[i-j]$ is defined as the matrix with elements n, m given by

$$R_{n,m}[i-j] = \int_{-\infty}^{\infty} \psi_k(n - iT_s - \tau_n)\psi_m^*(t - jT_s - \tau_m)\,\mathrm{d}t. \tag{2.15}$$

It is convenient to group the i^{th} terms in (2.13) to obtain a vector form of the equation

$$\boldsymbol{y}[i] = \boldsymbol{R^T}[1]\boldsymbol{A}\boldsymbol{\alpha}[i+1] + \boldsymbol{R}[0]\boldsymbol{A_m}\boldsymbol{\alpha}[i] + \boldsymbol{R}[1]\boldsymbol{A_m}\boldsymbol{\alpha}[i-1] + \boldsymbol{n}[i], \tag{2.16}$$

where $\boldsymbol{A_m}$ is a diagonal matrix containing the signal amplitudes, $\boldsymbol{A_m} = \mathrm{diag}\{A_1, \ldots, A_k\}$. $\boldsymbol{R}[0]$ is the cross-correlation matrix between the j^{th} and k^{th} signal for the current symbol, and $\boldsymbol{R_{jk}}[0] = \int_0^{T_s} \psi_k(t)\psi_j(t-(\tau_j-\tau_k))\,\mathrm{d}t$. The matrix $\boldsymbol{R}[1]$ contains the cross-correlations

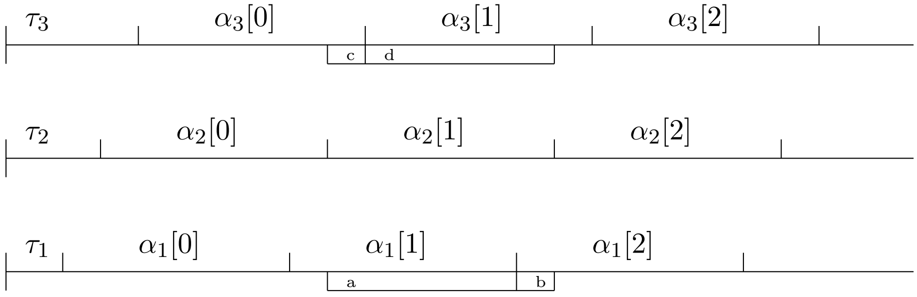


Figure 2.1: Three Asynchronous Cochannel Signals

between the j^{th} signal at the current symbol and the k^{th} signal at the next symbol for $j < k$

$$\boldsymbol{R_{jk}}[1] = \begin{cases} \mathbf{0}, & \text{if} j \geq k \\ \alpha_j[i-1] \int_0^{T_s} \psi_k(t)\psi_j(t + T_s - (\tau_j - \tau_k))\, \mathrm{d}t & \text{if} j < k \end{cases}. \tag{2.17}$$

For equally likely symbols in AWGN the MAP receiver is equivalent to the ML receiver that selects the symbol matrix $\hat{\boldsymbol{A}}$ such that the following equation is satisfied

$$\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \int_{-\infty}^{\infty} |r(t) - \sum_k s_k(t, \boldsymbol{\alpha_k})|^2 \, \mathrm{d}t, \tag{2.18}$$

where the symbol matrix $\boldsymbol{A}$ is a $(N_s \times N)$ matrix with elements $\alpha_k[i] \in \{-1, 1\}$. Equation (2.18) can be expanded and the terms that do not depend on $\boldsymbol{\alpha_k}$ dropped, to show this is the equivalent to maximizing

$$\begin{aligned} \Omega(\boldsymbol{A}) &= 2\int_{-\infty}^{\infty} r(t) \sum_k s_k(t, \boldsymbol{\alpha_k})\, \mathrm{d}t - \int_{-\infty}^{\infty} \left| \sum_k s_k(t, \boldsymbol{\alpha_k}) \right|^2 \mathrm{d}t \\ &= 2\int_{-\infty}^{\infty} r(t) \sum_k s_k(t, \boldsymbol{\alpha_k})\, \mathrm{d}t - \int_{-\infty}^{\infty} \sum_{k'} \sum_k s_k(t, \boldsymbol{\alpha_k}) s_{k'}^*(t, \boldsymbol{\alpha_{k'}})\, \mathrm{d}t \\ &= 2\int_{-\infty}^{\infty} r(t) \sum_k s_k(t, \boldsymbol{\alpha_k})\, \mathrm{d}t - 2\int_{-\infty}^{\infty} \sum_{k'} \sum_{k=1}^{k'-1} \Re(s_k(t, \boldsymbol{\alpha_k}) s_{k'}^*(t, \boldsymbol{\alpha_{k'}}))\, \mathrm{d}t \\ &- \int_{-\infty}^{\infty} \sum_k |s_k(t, \boldsymbol{\alpha_k})|^2 \, \mathrm{d}t \end{aligned}. \tag{2.19}$$

Where the last step follows from using the conjugate symmetry of the autocorrelation matrix to write the double summation as twice the real part of the sum of the lower triangular plus the diagonal terms. Equation (2.11) can be used to expand the first term of (2.19) and (2.12) to write this in terms of the matched filter outputs

$$\begin{aligned} 2\int_{-\infty}^{\infty} r(t)\sum_k s_k(t,\boldsymbol{\alpha_k})\,\mathrm{d}t &= 2\int_{-\infty}^{\infty} r(t)\sum_k\sum_i A_k\alpha_k[i]\psi_k(t-iT_s-\tau_k)\,\mathrm{d}t \\ &= 2\sum_k\sum_i A_k\alpha_k[i]y_k[i] \end{aligned}\quad. \tag{2.20}$$

The second term can be expanded using (2.11) to get

$$\begin{aligned} &\int_{-\infty}^{\infty}\sum_{k'}\sum_k s_k(t,\boldsymbol{\alpha_k})s^*_{k'}(t,\boldsymbol{\alpha_{k'}})\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty}\sum_k\sum_{k'}\sum_i A_k\alpha_k[i]\psi_k(t-iT_s-\tau_k)\sum_j A_{k'}\alpha_{k'}[j]\psi^*_{k'}(t-jT_s-\tau_k)\,\mathrm{d}t. \\ &= \sum_k\sum_{k'}\sum_i\sum_j A_k\alpha_k[i]A_{k'}\alpha_{k'}[j]R_{k,k'}[i-j] \end{aligned} \tag{2.21}$$

Where the matrix $\boldsymbol{R}[i-j]$ is defined as in Equation 2.15. Equation (2.20) and (2.21) can be combined to obtain:

$$\Omega(\boldsymbol{A}) = 2\sum_k\sum_i A_k\alpha_k[i]y_k[i] - \sum_k\sum_{k'}\sum_i\sum_j A_k\alpha_k[i]A_{k'}\alpha_{k'}[j]R_{k,k'}[i-j] \tag{2.22}$$

The above can be written in vector format as

$$\Omega(\boldsymbol{A}) = 2\sum_i \boldsymbol{\alpha}^T[i]\boldsymbol{A_m}\boldsymbol{y}[i] - \sum_i\sum_j \boldsymbol{\alpha^T}[i]\boldsymbol{A_m}\boldsymbol{R}[i-j]\boldsymbol{A_m}\boldsymbol{\alpha}[j]. \tag{2.23}$$

The symbol matrix $\boldsymbol{A}$ is a $(N_s \times N)$ matrix with column vectors $\boldsymbol{\alpha_k}$. Maximizing $\Omega(\boldsymbol{A})$ is a combinatorial problem; find $\hat{\boldsymbol{A}}$ by trying all 2^{N_sN} permutations of $\boldsymbol{A}$. Using this method computation is exponential in both message length N and number of users, N_s, making it an unattractive solution.

The key to reducing the complexity of the solution is to recognize that $\boldsymbol{R}[i-j]$ is nonzero only for $|i-j| \leq 1$ due to the finite support of ψ_k. This allows us to write

$$\hat{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}}\sum_i \Re(\lambda_i) \tag{2.24}$$

where $\Re(\cdot)$ indicates the real part and λ_i is given by

$$\lambda_i = 2\boldsymbol{\alpha}^T[i]\boldsymbol{A_m y}[i] - \boldsymbol{\alpha^T}[i]\boldsymbol{A_m R}[0]\boldsymbol{A_m \alpha}[i] - 2\boldsymbol{\alpha^T}[i]\boldsymbol{A_m R}[1]\boldsymbol{A_m \alpha}[i-1] \tag{2.25}$$

The last term has a factor of two from the symmetry of $R_{jk}[1] = R_{kj}[-1]$. The $\alpha[i]\alpha[i+1]$ from the previous term is absorbed into the $\alpha[i]\alpha[i-1]$ from the current term.

Writing the maximization problem in the form of (2.24), where λ_i depends only on $\boldsymbol{\alpha[i]}$ and $\boldsymbol{\alpha[i-1]}$, allows us to remove the exponential dependence on N. This maximization may be found using the Viterbi algorithm with 2^{2N_s-1} states and N stages. If we were to form a vector $\boldsymbol{y'} = \begin{bmatrix} y_1[0] & y_2[0] & \ldots & y_{N_s}[0] & y_1[1] & \ldots y_{Ns}[N] \end{bmatrix}^T$ as we did in the previous section we could perform this maximization using the Viterbi algorithm with 2^{N_s-1} states and $N_s \times N$ stages.

G. MAP ALGORITHM

There are two common varieties of the MAP algorithm. Bahl, Cocke, Jelinek and Raviv (BCJR) [83] is a forward-backward recursion to calculate the *a posteriori* probabilities of the state transitions of a Markov source observed through a discrete memoryless channel. The BCJR algorithm is detailed in Appendix B. Abend and Fritchman developed an alternative algorithm [75] that does not require a backwards pass, making it suitable for data that is not in finite blocks or for large blocks with a time-varying channel [84]. Li, et al. [84] provides a thorough discussion of the two types of MAP algorithms. In addition, many authors have investigated efficient hardware implementations of the BCJR algorithm [85].

1. Reduced Search Techniques

The MAP algorithm has complexity that is exponential in the number of users and the ISI channel length. There has been some work on reduced search techniques to lower the computational complexity of the BCJR algorithm [13, 14, 86]. These techniques all involve ignoring working probabilities that fall below a selected threshold. The methods can be roughly placed into two categories. The "M-BCJR" algorithm selects the M largest components at each stage during each pass (forward or backward) through the trellis. The "T-BCJR" algorithm ignores components at each stage that are below some threshold T. These techniques are geared toward their use in Turbo decoding where the BCJR algorithm is used over multiple iterations. The major reduction in computational complexity is for later iterations.

Vithanage, et al., [87] develop a reduced complexity BCJR type algorithm based on propagation of expectation . The reduction in complexity is similar to that of the M-BCJR algorithm proposed in [13, 14], by selecting a discrete distribution with limited support. They term this algorithm EP-MBCJR.

H. CHAPTER SUMMARY

This chapter provides required background and a survey of previous work in the area of cochannel interference mitigation and multi-user detection. The next chapter describes the AIS signal in detail and provide some analysis of the low-entropy property of this signal.

THIS PAGE INTENTIONALLY LEFT BLANK

III. AUTOMATIC IDENTIFICATION SYSTEM (AIS)

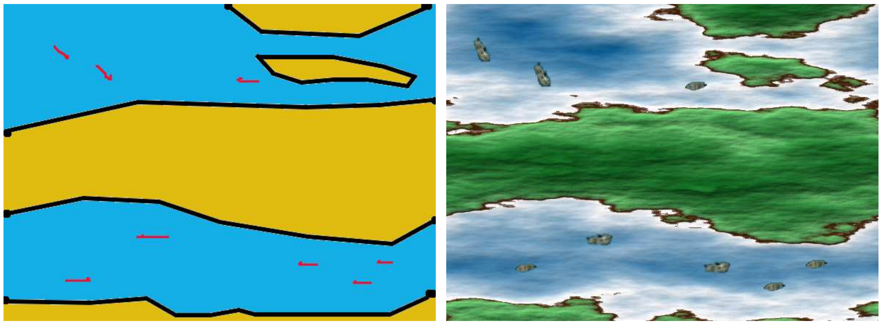
A scenario subject to potential cochannel interference is the reception of the ship-based AIS signal on a high-altitude platform such as an UAV or a satellite. There has been some interest in collection of the AIS signal from both high altitude UAVs and from satellites for maritime traffic monitoring [10, 16]. The AIS signal is particularly interesting for this dissertation due to its low-entropy property; the transmitted fields of the signal contain similar information from transmission-to-transmission. This property will be exploited to aid in joint detection; leading to improved FER performance.

A. AIS OVERVIEW

AIS is a ship- and land-based tracking and communications system operating in the VHF maritime band. The primary function of AIS is to provide information for surveillance and the safe navigation of ships [1, 4]. The AIS sends ship-based tracking messages indicating position and state information at intervals typically between ten and two seconds. The International Maritime Organization (IMO) has ruled that all passenger ships, cargo ships greater than 500 gross tons, and all ships greater than 300 gross tons on an international voyage must carry an AIS transceiver by July 2008 [88].

An example of the end use of an AIS transceiver is shown in Figure 3.1. Figure 3.1a shows an illustration of the output of a ship-board AIS display to aid in safe navigation. Figure 3.1b illustrates an aerial view of the same scene. The display provides the captain of a ship with a real-time picture of the local ship traffic around the vessel. The displays can be integrated with radar data to provide a integrated picture of local sea traffic.

Each AIS transmits position reports at a rate based on the current state of the ship. For example, a ship underway and changing course is required to report more often than a ship that is moored. Nominal reporting intervals for Class A and Class B equipment are given in Table 3.1 and Table 3.2 [1]. Class A Equipment is the mandatory equipment for ships required by the IMO to carry AIS equipment. Class B equipment is for ships not required by the IMO to carry equipment.



(a) AIS Display (b) Aerial View

Figure 3.1: AIS Display Example

Table 3.1: Class A Ship Reporting Interval

Class A Ships Information / Condition	Reporting Interval
Static Information	360 sec
Voyage Related Information	360 sec
Safety Message	as required
Ship at Anchor moving less than 3 knots	180 sec
Ship $<$ 14 knots not changing course	10 sec
Ship $14 - 23$ knots	6 sec
Ship $>$ 14 knots changing course	2 sec
Ship $>$ 23 knots	2 sec

Table 3.2: Class B Ship Reporting Interval

Class B Ships Information / Condition	Reporting Interval
Ship moving less than 2 knots	180 sec
Ship $2 - 14$ knots	30 sec
Ship $14 - 23$ knots	15 sec
Ship $>$ 23 knots	5 sec
Search and Rescue (airborne)	10 sec
Aids to Navigation	180 sec
AIS base station	10 sec

Table 3.3: AIS Transmitter Parameters

Transmit Parameter	Low Setting		High Setting	
Power (W)	2	± 20%	12.5	± 20%
Time-bandwidth product BT_s	0.3		0.4	
Modulation index h	0.25		0.5	
Bit rate	9600	± 50 ppm	9600	± 50 ppm
AIS 1 Frequency	161.975	± 3 ppm	161.975	± 3 ppm
AIS 2 Frequency	162.025	± 3 ppm	162.025	± 3 ppm

1. AIS Transmitter Characteristics

An AIS transmitter on the open seas will transmit a GMSK signal with a modulation index $h = 0.5$ and transmit power of 12.5 Watts. In terrestrial waters these settings may be changed by the appropriate authority [1]. Table 3.3 lists the various AIS transmitter configurable parameters [1]. On the open seas the “high” settings are used. In terrestrial waters an authority can instruct the transceiver to change any or all of the parameters to the “low” setting. The transceiver will default to alternate transmissions between the two AIS channels, AIS 1 and AIS 2.

2. AIS Multiple Access Method

The AIS is a multi-user system. The system allocates 2250 transmission time slots per minute on two separate frequencies allowing for a total of 4500 transmissions per minute. The AIS operating in continuous and autonomous mode uses SOTDMA to separate users. Each user within line-of-sight (nominally 55 km radius) will monitor AIS traffic for a full frame (1 minute) to create a list of candidate time slots for transmission. The user will then select randomly among these candidate time slots for transmission during the next frame. The same time slot will be used for up to seven subsequent transmissions. Each user effectively creates his own cell centered around his ship.

A minimal cochannel interference problem exists for users on the surface of the Earth; in general, users of the same time slot are not within line-of-sight. The time slot selection algorithm is designed such that the probability of two ships selecting the same time slot is low. A potential interference issue arises when receiving the signals on a platform that has a field-of-view larger than the nominal 110 km as illustrated in Figure 3.2. The three circles in the figure represent areas where the SOTDMA algorithm has separated the

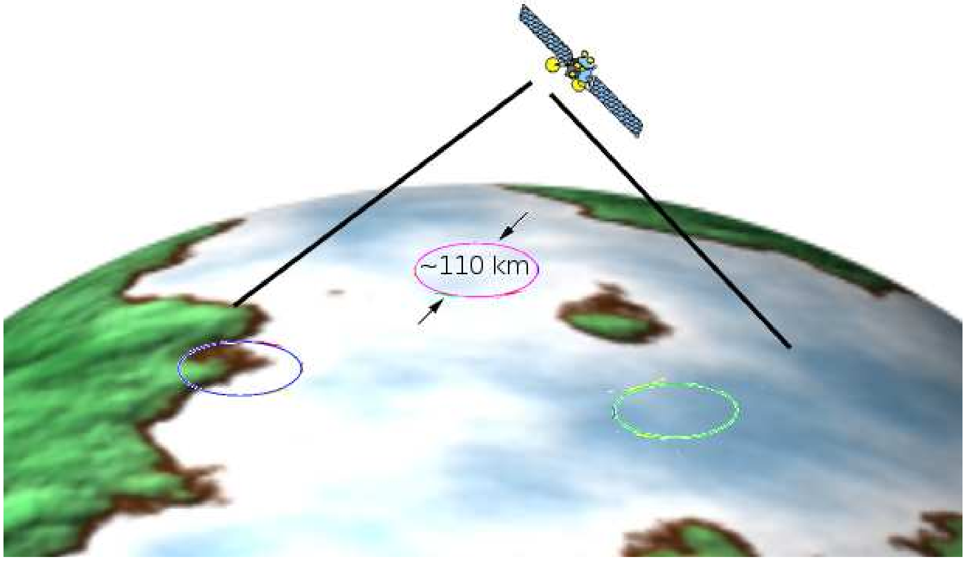


Figure 3.2: Example of Reception from Multiple Cells

signals via separate time slots. Within each region there is no contention; there is no more than one transmission per time slot per channel. At a remote receiver, like the satellite depicted in Figure 3.2, there is the possibility that some of the time slots will be occupied by multiple users leading to cochannel interference at the receiver.

The AIS does not directly address the hidden node problem. As an example of the hidden node problem, consider three ships Ship A, Ship B and Ship C all on a line and, separated by half the nominal cell size, Figure 3.3. Ship B can see both Ship A and Ship C but Ship A and Ship C can not see each other. If Ship A and Ship C choose the same time slot, then each will interfere with Ship B. This case does not cause a problem in practice. If the distance between ships is large, as is the required condition for the above interference, the importance of receiving the message is low.

The AIS specifies 2250 time slots each minute for each channel. Each slot is 26.67 ms (256 bits) long. The structure of the AIS message format is shown in Figure 3.4. This message structure is based on the High-level Data Link Control (HDLC) protocol specified in [89]. The "ramp up" period allows for the transmitter to get up to full power. The 24 bit training sequence is a sequence of alternating ones and zeros used for synchronization at the receiver. The start flag and the end flag are the HDLC flags indicating the start and end respectively of the 168-bit data packet. The frame check sequence (FCS) is a 16 bit

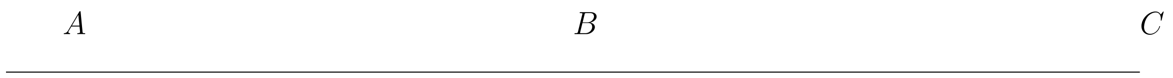


Figure 3.3: Hidden Node Example

Ramp up	Training sequence	Start flag	data	FCS	End flag	Buffer
8 bits	24 bits	8 bits	168 + stuff bits	16 bits	8 bits	24 bits

$26.67ms$

Figure 3.4: AIS Packet

cyclic redundancy check (CRC) on the data portion of the message to increase confidence that a message is received without error. The CRC uses the polynomial specified in [89] with bits preset to one. The buffer allows for timing offsets from bit-stuffing, propagation delay, repeater delay, and synchronization jitter.

Table 3.4 is a paraphrased description from the International Telecommunication Union (ITU) recommendation [1] of the contents of the data portion of the AIS packet described in Figure 3.4. There are many varieties of AIS messages. Table 3.4 illustrates some of the more common messages. Careful inspection of Table 3.4 reveals that many of the fields are either constant or predictable. The ability to predict field values is the key feature exploited in this work; this is explained in more detail in Section III.D.

B. EXAMPLE AIS LINK BUDGET AT LEO

It is useful to calculate the anticipated SNR at the receiver for an example collection. If the reception is severely noise-limited for single-channel reception, little will be gained from investigating methods to mitigate cochannel interference. An example AIS link budget calculation for a collector with a path length of 800 km is shown in Table 3.5. This calculation does not take into account factors such as multipath and considers a rather

Table 3.4: Type 1, 2 or 3 AIS data message break out after [1]

Parameter	**Number of bits**	**Description**
Message ID	6	1, 2, or 3 for this message
Repeat Indicator	2	Indicates number of times message has been repeated
User ID	30	MMSI[a] number
Navigation Status	4	0 under way, 1 at anchor, 2 not under command ...
Rate of Turn	8	$\in [-127, 127]$, -128 indicates not available
SOG	10	Speed over ground $\in [0, 1022]$ 1023 not available
Position accuracy	1	1 = high (< 10 m)
Longitude	28	`0x6791AC0` indicates not available
Latitude	27	`0x2412140` indicates not available
COG	12	Course over ground, tenths of a degree $\in [0, 3600]$, `0xE10` indicates not available
True heading	9	Degrees $\in [0, 359]$, 511 indicates not available
Time stamp	6	UTC[b] seconds $\in [0, 59]$, codes $\in [60, 63]$
Reserved	4	Reserved
Spare	1	should be set to 0
RAIM-flag	1	RAIM (Receiver autonomous integrity monitoring), 1= RAIM in use
Communications state	19	Specifies slot position and other TDMA parameters
Total	168	

[a] Maritime Mobile Service Identity (MMSI)

[b] universal time (UTC)

simple gain for the transmit and receive antennas. The path loss is calculated as

$$\text{Path Loss} = \left(\frac{4\pi R}{\lambda}\right)^2 \tag{3.1}$$

where R is the range and λ is the carrier wavelength given by 3×10^8(m/s)/162(MHz). The received power is given by

$$\text{Received Power} = P_t G_t G_r / \text{Path Loss} \tag{3.2}$$

where P_t is the transmitted power and G_t and G_r are the transmit and receive antenna gains respectively. The energy per bit (E_b) is given by the Received Power $\times T_s$ where T_s is the bit period. The one sided noise density N_0 is given by $N_0 = kT$ where k is Boltzman's constant and T is the noise temperature.

Table 3.5 demonstrates that for nominal reception from a high altitude receiver, the signal is not noise-limited, therefore there may be opportunities for joint reception techniques.

Table 3.5: Example AIS Link Budget

Parameter		Val	dB
Path Length	R	800.00 (km)	
Transmit Frequency	f	162.00 (MHz)	
Path Loss	L_s		134.69 (dB)
Path Loss	L_s		134.69 (dB)
Power Transmitted	P_t	12.50 (W)	10.97 (dBW)
Transmit Antenna Gain	G_t	1.00	0.00 (dB)
Receive Antenna Gain	G_r	1.00	0.00 (dB)
Received Power	P_r	4.24^{-13} (W)	-123.72 (dBW)
Received Power	P_r	4.24^{-13} (W)	-123.72 (dBW)
Bit Period	T_b	0.10 (mSec)	-39.82 (dB-sec)
Energy per Bit	E_b	4.42^{-17} (J)	-163.55 (dBJ)
Representative Noise Temperature	T	2910.00 Kelvin	34.64 (dBK)
Boltzman Constant	k	1.38^{-23} JK^{-1}	-228.60 (dBJK^{-1})
Two Sided Noise Density	N_0		-193.96 (dBJ)
Energy per Bit	E_b		-163.55 (dBJ)
Two Sided Noise Density	N_0		-193.96 (dBJ)
Energy per Bit / Noise Density	E_b/N_0	1100.22	30.41 (dB)
Received Power	P_r	4.24^{-13} (W)	-123.72 (dBW)
Noise Density	N_0		-193.96 (dBJ)
Bandwidth	B	25.00 (kHz)	43.98 (dB)
Signal to Noise Ratio	SNR	422.48	26.26 (dB)

C. SINGLE-CHANNEL DEMODULATION

Before delving too far into demodulation of cochannel signals, it is instructive to consider a simple case of non-coherently demodulating an AIS packet in the presence of AWGN. This example, although not optimal in any sense, illustrates the characteristics of the signal. The test signal has the following characteristics:

- data rate of 9600 baud
- center frequency of $\approx$ 160 MHz (AIS1 161.975 MHz, AIS2 162.025 MHz)
- GMSK modulated with $BT_s = 0.4$, $h = .5$
- Frames on UTC minute boundary with 2250 time slots
- Each slot 26.6 ms
- Before modulation, the data are encoded non-return-to-zero inverted (NRZI) (level changes when a '0' is sent, level stays the same when a '1' is sent)[1].
- The data field and FCS are bit-stuffed (when a string of five consecutive 1's is encountered a '0' is inserted).
- A 24 bit training sequence of alternating 0's and 1's. It is not specified if the pattern starts with 0's or 1's since the data will be NRZI encoded.
- The data contains the HDLC start and stop flags 0x7E, where the 0x notation indicates a hexadecimal value.

Assume the signal has been received through a band-pass filter and mixed down in quadrature so the signal is analytic and centered at some intermediate frequency (IF). The IF is then sampled at some rate greater than twice the signal bandwidth. The first step in demodulation is to identify the start of the packet in time. An energy detection routine can be used to do this. Figure 3.5 is a time frequency plot of a single AIS packet in noise located by using an energy detection method.

[1] The definition of NRZI is found in many sources defined as a level change when a '1' is sent and no change on a '0'. The definition above for the AIS is from the ITU recommendation [1].

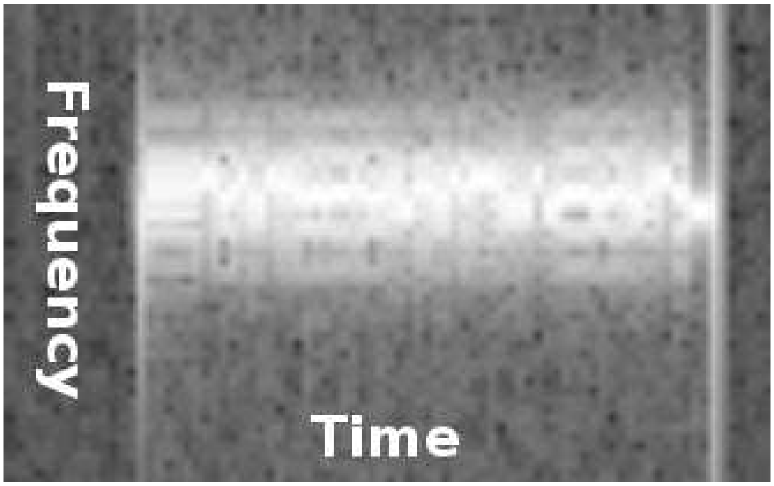


Figure 3.5: Single AIS Signal in AWGN

The next step is to pass the signal through a band-pass filter, removing out-of-band noise, and then shift the result down to baseband. Let the received analytic sequence be represented by

$$r[n] = s[n] + z[n]. \tag{3.3}$$

The result from filtering and mixing down to baseband is

$$y[n] = (r[n] * h[n])\, e^{-j\omega_c n}, \tag{3.4}$$

where ω_c is the approximate intermediate frequency and $h[n]$ is the impulse response of the filter. The differential phase term, $\Delta\phi$, is given by

$$\Delta\phi[n] = \phi[n] - \phi[n-1], \tag{3.5}$$

where

$$\phi[n] = \tan^{-1}\left[\frac{\Im(y[n])}{\Re(y[n])}\right], \tag{3.6}$$

and $\tan^{-1}(\cdot)$ is the four-quadrant arctan function. Figure 3.6 shows $\Delta\phi[n]$ for an example received AIS signal. The training sequence of alternating ones and zeros can clearly be seen. Notice that the alternating ones and zeros show up as alternating pairs of ones and pairs of zeros from the NRZI encoding. For example the sequence "010101010101" will be

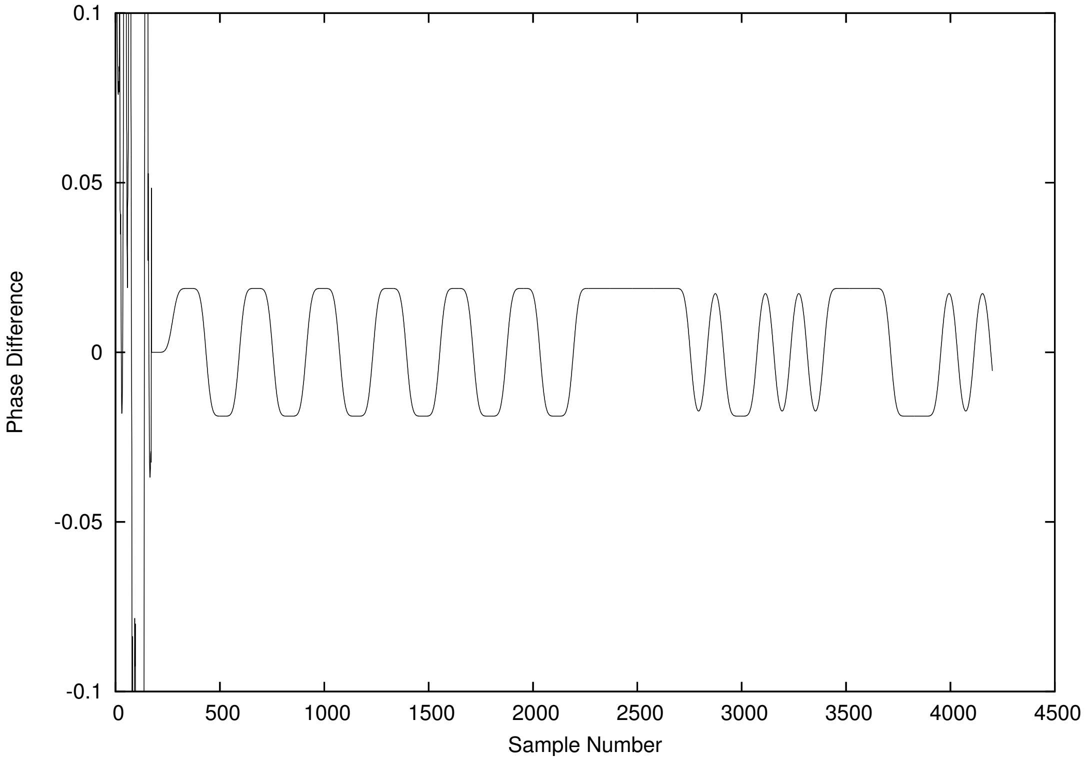


Figure 3.6: Differential Phase

either "001100110011" or "110011001100" after NRZI encoding, depending on the initial state of the encoder.

The AIS signal has a training sequence of alternating 1's and 0's followed by the HDLC start flag of 0x7E. The combination of these sequences can be used to locate the start of the packet and to derive bit timing.

A data sequence is then generated by decimating $\Delta\phi[n]$ with an offset and making hard decisions.

$$a_{nrzi}[n] = \mathrm{u}(\Delta\phi[Dn + k] - \overline{\boldsymbol{\Delta\phi}}) \tag{3.7}$$

where D is the down-sampling factor, k is an integer offset, $\overline{\boldsymbol{\Delta\phi}}$ is the mean of $\boldsymbol{\Delta\phi}$, and $\mathrm{u}(\cdot)$ is the unit step function.

$$u(\zeta) = \begin{cases} 1 & \zeta \geq 0 \\ 0 & \zeta < 0 \end{cases} \tag{3.8}$$

Note that Equation 3.7 assumes down-sampling by an integer. If the end goal is to recover symbols at some rate R_s, the sample rate should be fixed at $F_s = lR_s$ for some positive integer l. If this were not the case two options exist: re-sample the original data such that $F_s' = lR_s$ or select the sample which is closest to the ideal sample time.

Since the data $a_{nrzi}[n]$ is encoded NRZI it must be converted to non-return-to-zero (NRZ) defined by:

$$a_s[n] = a_{nrzi}[n] \oplus a_{nrzi}[n-1], \tag{3.9}$$

where $\oplus$ denotes the logical XNOR operation. Before the data are transmitted it is bit-stuffed. The bit-stuffing operation inserts a zero after a consecutive string of k ones. This has a benefit of aiding timing recovery and removing the ambiguity with detection of control flags. The data recovery operation is written as an operator B_5^{-1}, for example

$$\boldsymbol{a} = B_5^{-1}(\boldsymbol{a_s}). \tag{3.10}$$

Where $\boldsymbol{a_s}$ is the bit-stuffed vector, $\boldsymbol{a}$ is the recovered original data vector prior to bit-stuffing. The operator B_5^{-1} indicates the unstuffing operation when $k = 5$, as is the case for AIS.

Data are sent byte-wise with the least significant bit first. The data are then in a format ready to be parsed to recover the internal data fields. A full description of all packet formats is given in [1].

D. PREDICTION OF AIS PARAMETERS

Suppose f is a discrete random variable representing a number coded with k bits. If the probability mass function (PMF) of f is known to the receiver, the receiver may use this information in deciding which of the possible values, $\{0, 1, \ldots, 2^{k-1}\}$, of f were sent. When the PMF is not uniform on $[0, 2^{k-1}]$ there will be some gain in performance by using MAP detection vs. ML detection. The details of the gain in performance are provided in Chapter VI and Chapter VII. The predictability of various discrete random variables in AIS data fields are discussed below.

1. Prediction of an AIS Packet Time-of-Arrival

Time-of-arrival prediction is broken into two estimates. The first is an estimate of which time slot a particular AIS transceiver will transmit on. This estimate is based on the SOTDMA time slot selection algorithm. The second is an estimate of the offset from the

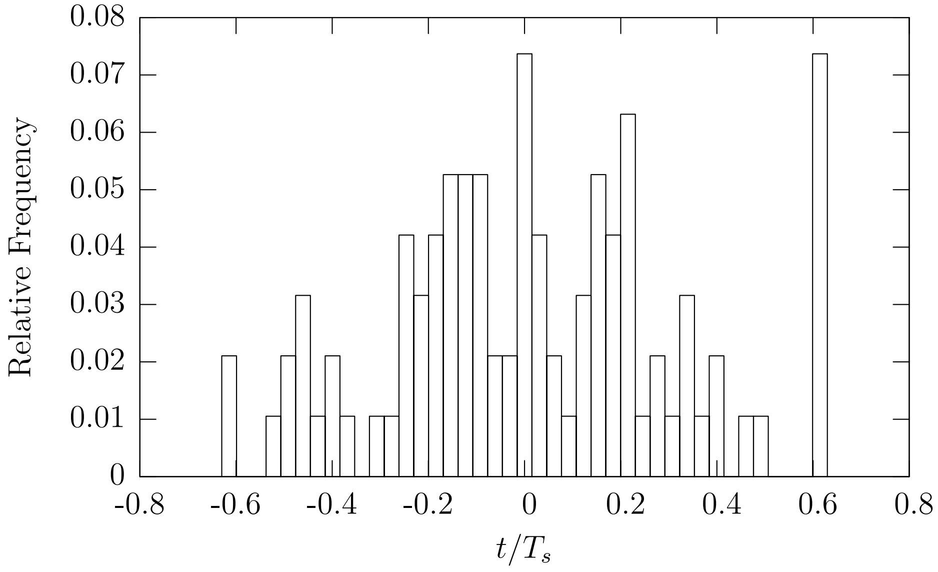


Figure 3.7: Histogram of Time-of-Arrival Variation for Several AIS Transceivers

nominal start of that time slot. This estimate is based on the nominal path length to the receiver and an estimation of the stability of the transmissions from the AIS transceivers.

A particular AIS transmitter will reserve a fixed number of time slots in a SOTDMA frame depending on the desired transmit rate. Typically between 6 and 30 time slots per frame will be reserved, corresponding to update rates of ten and two seconds respectively. It is important to predict with some degree of accuracy on which time slot a particular user will transmit. This is particularly important for identifying the start of a packet in a cochannel reception.

The same time slot will be used for up to seven frames before the next assignment. The number of frames for which a time slot is reused is determined by the SOTDMA algorithm. Chapter IV describes the time slot assignment algorithm in great detail.

The specification in the ITU recommendation [1] allows for ±3 symbols of jitter for the packet start time relative to the nominal start time. Experimental results have shown that under typical conditions this jitter interval is on the order of 3-100 μs; this corresponds to 3-25% of a symbol period. Figure 3.7 shows the distribution of time-of-arrival offsets for several transceivers.

2. Prediction of AIS Packet Data

We may ask, "How much information is transmitted by an AIS message?" To answer this question the information source is modeled as a stochastic random process. The entropy of an error term can then be calculated, where the error term is the difference between the measured value and the value modeled by a random process. Most of the fields within the data portion of an AIS message are very predictable. It is instructive to analyze each field of one of the message types and present results for each.

Analysis of data received by the United States Coast Guard (USCG) shows that type 1, 2 and 3 messages make up approximately 98% of AIS message traffic. Analysis was performed by obtaining 24 hours of coastal data from the USCG-provided application AISUser and counting the number of each message type. With this in mind the analysis in this work is restricted to these message types; other message types are treated as unknown interferers.

Four types of predictors are used depending on the field. The latitude and longitude fields are predicted with a Kalman filter. The communications state field uses a special predictor based on the protocol for SOTDMA. The time field uses an external time source as a predictor. All of the other fields use a first order predictor. A full description of each predictor along with the parameters used for each field is provided in Chapter IV.

Some of the fields vary depending on the particular AIS transceiver used, and this warrants additional explanation. For example, the latitude and longitude fields are 27 and 28 bit respectively with the least significant bit corresponding to 1/10,000 minute. Some AIS report to 1/1,000 minute or even 1/100 minute, however this is taken into account by the predictor developed in the next chapter.

Each field is modeled as a stochastic process $f(t)$. For each field past measurements at time $t - \tau_k$ are used to estimate the current value at time t.

$$\hat{f}(t) = g(f(t-\tau_1), f(t-\tau_2), \ldots, f(t-\tau_N), \tau_1, \tau_2, \ldots, \tau_N) \tag{3.11}$$

The random process $f(t)$ is discrete; the possible values of $f(t)$ are drawn from a finite set.

In order to measure the entropy of the probability distribution of the error in the prediction of $\hat{f}(t)$, the function $e(t)$ is defined as the difference between the estimated value and the actual value.

$$e(t) = \hat{f}(t) - f(t) \tag{3.12}$$

Entropy is a natural measure of uncertainty in the random process. The entropy of the random process $f(t)$ could be considered, but it is clearly non-stationary. Although $e(t)$ lacks true stationarity, $f(t)$ could not even be approximated as a stationary process.

The entropy (in bits) of a discrete random variable $\boldsymbol{p}$ with probability mass function $(p_1, p_2, \ldots, p_K)$ is given by:

$$H(\boldsymbol{p}) = \sum_{k=1}^{K} p_k \log_2(1/p_k), \tag{3.13}$$

where we define $0 \log_2(1/0) = 0$.

To find the entropy of $e(t)$ the PMF of $e(t)$ is required. The PMF is estimated using a normalized histogram, implicitly assuming that the process $e(t)$ is ergodic. With the PMF of $e(t)$, Equation (3.13) can be used to calculate the entropy (information content) of the field.

Figure 3.8 shows the number of bits in each field of an AIS type 1, 2, or 3 message, along with the calculated entropy of the residual error process $e(t)$. The message data used to create this graph is from the USCG collection platforms along the US coastline.

E. CHAPTER SUMMARY

This chapter introduces the AIS signal as a good candidate for low-entropy cochannel mitigation techniques. The characteristics of the transmitter are introduced and multiple access scheme SOTDMA is discussed. The case for cochannel interference as seen by a remote observer is presented. Much detail is provided on the format of the transmitted messages and the data contained within each message. Finally measured results for the estimated entropy of the transmitted data is presented, illustrating the AIS signal is low-entropy. The next chapter discusses the specific methods used for prediction of the AIS messages for this research.

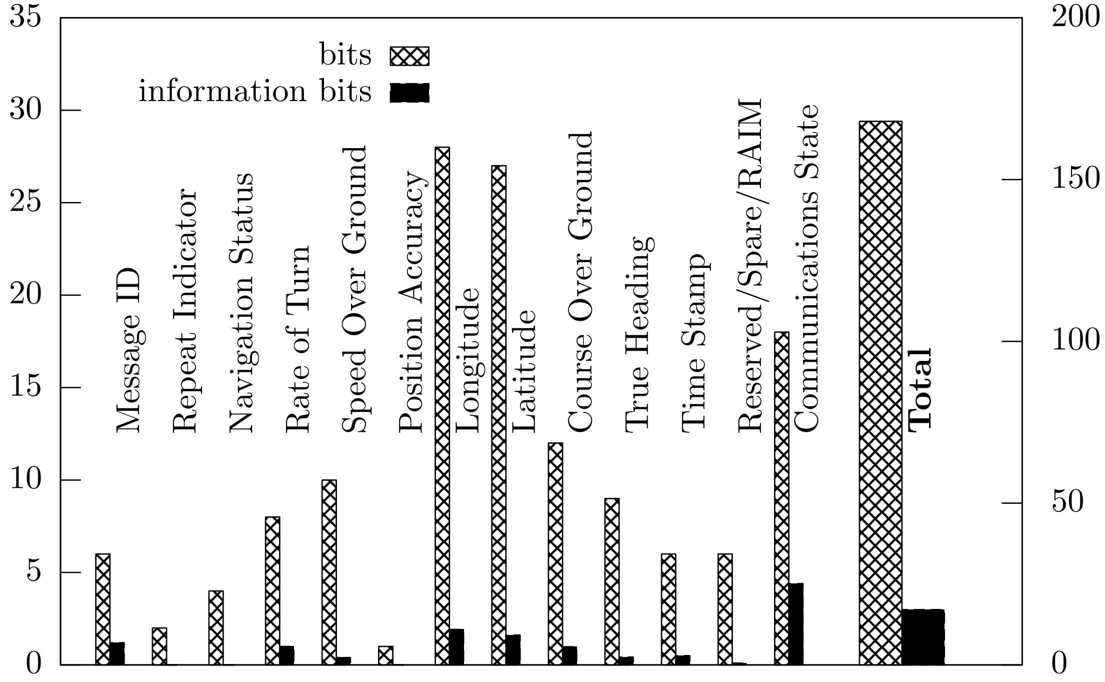


Figure 3.8: Comparison of Transmitted Bits and Entropy of Residual Error on Field by Field Basis

IV. PREDICTION

This dissertation develops an algorithm for joint detection of two or more signals exploiting *a priori* knowledge of the data source for one or more of the signals. The fundamental assumption is that we are able to predict in some way future values of the desired signals, and use these predicted values to aid in demodulation. A distribution is desired for all possible values of a signal based on past (prediction), and perhaps both past and future (smoothing) measurements. This chapter introduces some methods of prediction and applies them to the AIS signal. Results are then presented for the application of these methods to a 24 hour collection of the AIS signals on a selected set of ships.

Before discussing prediction, it is necessary to describe some of the characteristics of the values that are to be predicted. These values are *reported* as discrete values. For some values, such as position, the underlying value is continuous, but the data itself is a quantized version of that continuous value. For other fields, such as the AIS field msg_id, the underlying value as well as the measurement are discrete. Different predictors are used for each of these underlying models.

For each field, the process noise describes the deviation of the value predicted by the model from the actual value. The measurement noise describes the error in measuring the actual value. The auto- and cross-correlation functions of the model (process) noise or the measurement noise are not known; they are estimated.

A third and very important point is that the error term is not the difference between the true state and the estimate. The error term is the difference between the estimate and the reported, or quantized state. This is because the estimate is used to help decide what was transmitted, and it is the quantized values that are transmitted. It is also interesting to note that if the underlying noise in the system is significantly less than the quantization error, then the error in the estimate can not accurately be modeled as a stationary random process. Depending on how the true distribution aligns with the quantization levels, the PMF of the quantized value could be quite different from that of the true distribution. To see this, consider a quantized random walk. When the underlying true value is close to a quantization level, the output will likely be constant. When the true value is exactly between quantization levels, the output will alternate between the two levels.

Prediction is a tool used in this work, but not the focus of the work. The development below is intended to introduce the concepts to support their later use as tools.

The objective of this chapter is not to develop the "best" predictor. The objective is to demonstrate that the prediction assumed in the rest of the dissertation is possible, and to quantify the performance of the predictor.

A. FIRST ORDER PREDICTOR

The least complicated of the predictors considered is the first order predictor. The first order predictor estimates the value at time $n+l$ as a factor times the value at time n [90]:

$$f[n+l] = a_0 f[n] \tag{4.1}$$

For many applications, the first order predictor with $a_0 = 1$ may be the most appropriate predictor. Consider a remote outdoor temperature sensor with a resolution of 1/2 degree Celsius reporting once per second. A practical predictor may suggest future values are identical to current values. The first order predictor is appropriate when the process is modeled as discrete Markov, and the probability of transition out of the current state is less than the probability of remaining in the current state. This occurs when the diagonal elements of the transition matrix are greater than any other element in a particular row, $A_{i,i} > A_{i,j} \quad \forall j \neq i$. The AIS Navigation Status field is an example of a process that can be accurately modeled as a discrete Markov process with this property. The primary advantage of the first order predictor, and the reason it is used here, is simplicity.

Most of the fields in the type 1, 2, or 3 AIS messages are modeled as discrete Markov chains with the property described above. For the type 1, 2 or 3 AIS packets a first order predictor is used for the following fields: Repeat Indicator, User ID, Navigation Status, Rate of Turn, Speed Over Ground, Position Accuracy, Course over Ground, and

True Heading. The residual entropy of the error from a first order predictor for these AIS type 1, 2 and 3 fields are given in Figure 3.8 for an example 24 hour data set.

B. SECOND ORDER PREDICTOR

The second order predictor is another very simple predictor. It predicts the next value based on the current value and a previous value using linear interpolation [90].

$$f[n+l] = f[n] + l\frac{f[n] - f[n-k]}{k} = \sum_{i=0}^{1} a_i f[n-i] \tag{4.2}$$

The second order predictor is also very simple to implement. The real problem with the second order predictor is that noise on the measurement can be magnified by the l/k factor. Because of this limitation, the second order predictor is not used for any of the AIS fields.

A better predictor is one that somehow weights new measurements based on some knowledge of how much noise is in the system. Predictors based on the Kalman filter have this characteristic. The Kalman filter is introduced in the following section.

C. KALMAN FILTER

The Kalman Filter is a recursive filter that provides an optimum, MMSE estimate of an unobserved state, provided that the system can be modeled as a Markov chain with linear operators relating the next state to the current state, perturbed by white noise [91]. In this work, the Kalman filter is used as a predictor to estimate AIS fields that nominally change at a constant rate, such as latitude and longitude. The Kalman filter is described below using an example from [91] and [92].

Consider predicting the one-dimensional position of a train on a straight track. The unobserved state consists of position and velocity. The system is modeled as ideal; having no external forces leads to a state transition equation of

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{w}_k, \tag{4.3}$$

where $\boldsymbol{\Phi}_k$ is a 2×2 state transition matrix, k is the time index, and $\boldsymbol{x}$ is the column state vector with first and second elements of position and velocity, respectively. The process noise vector, $\boldsymbol{w}_k$, is a zero-mean multivariate Gaussian random variable with covariance matrix $\boldsymbol{Q}_k = \mathcal{E}\{\boldsymbol{w}_k\boldsymbol{w}_k^T\}$. The vector $\boldsymbol{w}_k$ is uncorrelated in time, $\mathcal{E}\{\boldsymbol{w}_k\boldsymbol{w}_i^T\} = \boldsymbol{0},\ \forall i \neq k$. The noise vector $\boldsymbol{w}_k$ represents the deviation from the ideal model due to unmodeled forces. For the train example the deviation from the model could be caused by a slight incline or uneveness of the track, the conductor accelerating, etc. All of these effects are modeled with the noise term $\boldsymbol{w}_k$.

For this example, the transition matrix is:

$$\boldsymbol{\Phi}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{4.4}$$

Where Δt is the time step between successive measurements. Position at time $k+1$ is the position at time k plus Δt times the velocity term, and the velocity at time $k+1$ is equal to velocity at time k.

The observation at time k is denoted by the vector $\boldsymbol{z_k}$. The mapping of the true value of the state $\boldsymbol{x}_k$ to our measurement $\boldsymbol{z}_k$ is through the matrix $\boldsymbol{H}$

$$\boldsymbol{z_k} = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{4.5}$$

where the measurement noise vector $\boldsymbol{v}_k$ is a zero-mean multivariate Gaussian random variable with covariance matrix $\boldsymbol{R}_k = \mathcal{E}\{\boldsymbol{v}_k\boldsymbol{v}_k^T\}$. As in the case of the process noise the vector $\boldsymbol{v}_k$ is uncorrelated in time, $\mathcal{E}\{\boldsymbol{v}_k\boldsymbol{v}_i^T\} = \boldsymbol{0},\ \forall i \neq k$.

For this example assume position is the only measurable quantity. The measurement vector $\boldsymbol{z}_k$ is then a scalar, and the matrix $\boldsymbol{H}$ reduces to a row vector.

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.6}$$

The algorithm starts with an initial state estimate, $\hat{\boldsymbol{x}}_0$, and covariance matrix, $\boldsymbol{P}_0$, of the initial estimate, where $\boldsymbol{P}_k = \mathcal{E}\{(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T\}$. The algorithm then follows a cycle of predicting ahead using the transition matrix followed by updating the prediction with a measurement.

The estimate of the next state, prior to measurement, is found using the previous estimate of the state and the transition matrix to project ahead:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k}, \tag{4.7}$$

where the notation $\hat{\boldsymbol{x}}_{j|k}$ indicates the estimate $\hat{\boldsymbol{x}}$ at time j based on all the information up to time k.

The estimate of the error covariance is also projected ahead using:

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{\Phi}_k \boldsymbol{P}_{k|k} \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{4.8}$$

The estimate is then updated with the current measurement as:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k-1}) \tag{4.9}$$

Where $\boldsymbol{K}_k$ is the *Kalman gain* (described below) and the term in the parentheses, $\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k-1}$, represents the new information present in the measurement.

The estimate of the error covariance is also updated with the new gain

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} \tag{4.10}$$

where $\boldsymbol{I}$ is the identity matrix.

The Kalman gain, $\boldsymbol{K}_k$, is selected to minimize the mean square error $\mathcal{E}\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^T(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})\}$. The mean square error for the estimate is the trace of the matrix $\boldsymbol{P}_k$. It can be shown [91], that the Kalman gain that minimizes the mean square error is given by:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^t (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^t + \boldsymbol{R}_k)^{-1} \tag{4.11}$$

Inspection of Equation (4.11) reveals that when the elements of $\boldsymbol{R}_k$ are large, indicating large uncertainty in the measurements, the elements of the Kalman gain are small in magnitude. This effectively lowers the weight of new measurements in updating the state in Equation (4.9) and updating the estimate of the error covariance in Equation (4.10). The next state would then be determined almost entirely by the previous state and the state transition matrix as given in Equation (4.7). A trade-off exists in selecting $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ between convergence and how well the filter attenuates noise. If the filter is modeled with too little process noise $\boldsymbol{Q}_k$ or too large measurement noise $\boldsymbol{R}_k$, the filter will take a very long time to converge, or may never converge, but will do a very good job of attenuating noise in the measurements. On the other hand, if the filter is modeled with too large process noise $\boldsymbol{Q}_k$ or too little measurement noise $\boldsymbol{R}_k$, the filter will converge rapidly but do a poor job of attenuating measurement noise.

Table 4.1: Prediction algorithm based on the Kalman filter

```
$\hat{\boldsymbol{x}}_0 \Leftarrow$ initial estimate
$\hat{\boldsymbol{P}}_0 \Leftarrow$ initial estimate
loop
    $\boldsymbol{K}_k \Leftarrow \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^t(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^t + \boldsymbol{R}_k)^{-1}$    {Compute Gain}
    $\hat{\boldsymbol{x}}_{k|k} \Leftarrow \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1})$    {Update State Estimate}
    $\boldsymbol{P}_{k|k} \Leftarrow (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}$    {Update Estimate Error Covariance}
    $\hat{\boldsymbol{x}}_{k+1|k} \Leftarrow \boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k}$    {Project ahead to next state}
    $\boldsymbol{P}_{k+1|k} \Leftarrow \boldsymbol{\Phi}_k\boldsymbol{P}_{k|k}\boldsymbol{\Phi}_k^t + \boldsymbol{Q}_k$    {Project ahead Estimate of error covariance}
end loop
```

If the parameters for the Kalman filter are known, the Kalman filter provides the optimum MMSE estimate of the state [91]. The steps in the Kalman filter algorithm are summarized in Table 4.1.

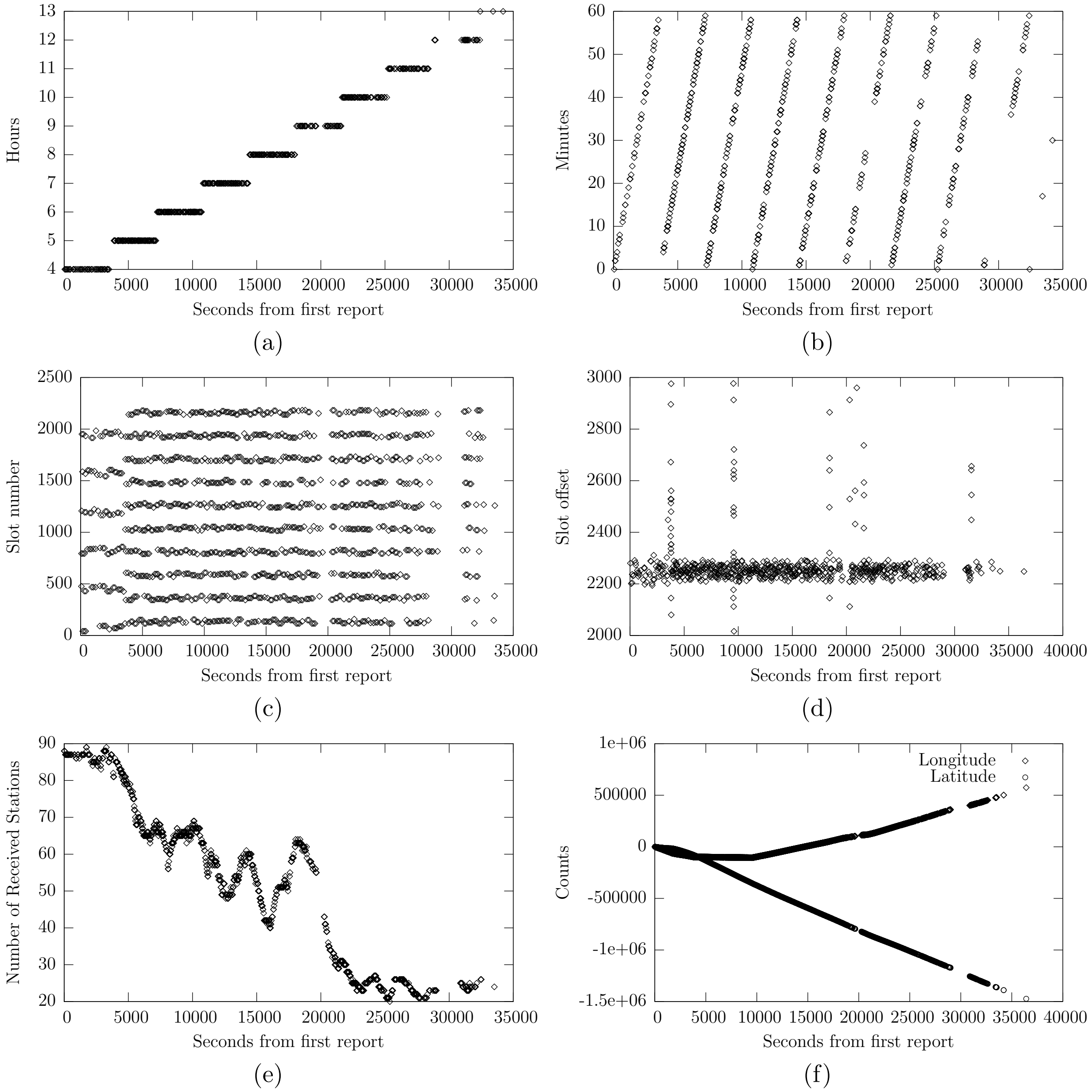
D. SPECIAL NETWORK FLAG PREDICTOR

The SOTDMA Communication State field is structured as shown in Table 4.2 and Table 4.3 [1]. Clearly, most of the parameters in the table are predictable once the first frame has been received. Some of the parameters are effectively deterministic, such as time.

The SOTDMA Communication State field of an AIS message is almost entirely predictable. Figure 4.1 is presented to give an example of typical SOTDMA Communication State field values as a function of time. The figures were created using data collected from a single ship over a 24 hour period. Of the possible values contained in the Communication State field, only the number of received stations and the time slot offset are not deterministic.

A difficult problem in determining the value of the communication state is determining the “timeout” value. When the timeout reaches zero, a new value is selected randomly by the SOTDMA algorithm. The probability of correctly guessing this new value is 1/7. Unfortunately, this timeout value also determines the meaning of the submessage.

The algorithm developed for predicting the Communication State field is summarized in Table 4.4. The algorithm begins with the time of the message reception in seconds of day (SOD), the received data (rx), and the previous values: timeout, SOD, number of



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.1: SOTDMA fields

Table 4.2: SOTDMA communication state after [1]

Parameter	Bit Position	Description	Parameter Modeled As
Sync State	19:18	0 UTC Direct, 1 UTC indirect, 2 Station synchronized to another base station	random constant
slot time-out	17:15	Specifies the number of frames remaining until a new slot is selected, 0 indicates this is the last transmission in this slot	Random constant and deterministic
Sub Message	14:1	Submessage meaning depends on value of the timeout as described in Table 4.3	See Table 4.3

Table 4.3: SOTDMA communication state submessage after [1]

Slot time-out	Submessage	Description	Parameter modeled as
3,5,7	Received Stations	Indicates the number of other stations currently received (between 0 and 16383)	slowly varying random walk
2,4,6	slot number	Slot number used for this transmission (between 0 and 2249)	deterministic
1	UTC hour and minute	Hours is indicated in bits 13:9 minute is coded in bits 8:2	approximately deterministic
0	slot offset	Indicates the relative jump to the time slot for transmission in the next frame.	Uniform distribution around current slot number

received stations, and slot number. It first generates estimates based on prior received values, then demodulates the received data based on these estimates. Finally, it updates the stored values to be used for future predictions.

E. RESULTS FOR AIS

In this section some experimental results are presented for the prediction algorithms developed for the AIS field data. A 24 hour data set provided by the USCG and extracted

Table 4.4: Predict communication state

```
$i \Leftarrow sod \mod 60$
                                                    {predictions}
$estTimeout \Leftarrow (pTimeout[i] - \lfloor (sod - pSod[i])/60 + 0.5 \rfloor) \mod 7$
$estSyncState \Leftarrow pSyncState$
if $estTimeout = 3|5|7$ then
    $estSubMessage \Leftarrow pSlot[i]$;
else if $estTimeout = 2|4|6$ then
    $estSubMessage \Leftarrow numStations$;
else if $estTimeout = 1$ then
    $estSubMessage \Leftarrow \lfloor (sod \mod 24*60*60)/(60*60) \rfloor * 2^9 + \lfloor (sod \mod 60*60)/60*2^2 \rfloor$
else if $estTimeout = 0$ then
    $estSubMessage \Leftarrow 2248$
end if
                                                    {demodulate}
$\{timeout, syncState, sub\} \Leftarrow demod(r, est)$
                                                    {update for future predictions}
$pSod[i] \Leftarrow sod$
$ptimeout[i] \Leftarrow timeout$
$psyncState[i] \Leftarrow syncState$
if $timeout = 3|5|7$ then
    $pSlot[i] \Leftarrow subMessage$
else if $estTimeout = 2|4|6$ then
    $numStations \Leftarrow subMessage$
else if $estTimeout = 0$ then
    $newSlot \Leftarrow (slot[i] + subMessage - 2250) \mod 2250$
    $pTimeout \Leftarrow 7$
end if
```

with the AISUser application is used to generate the results presented in this section. The receptions are sorted by transmitting ship using the MMSI field. The data is then passed to a predictor that attempts to predict the next message field-by-field given the previous messages.

The receive station that generated this data received reports from 192 unique ships on the collection date. Results from three of the 192 ships are presented below.

Three ships were selected for analysis. Two ships under way with two different reporting precisions for the latitude and longitude fields. Ship A has a reporting precision

100 counts and Ship C has a reporting precision of 10 counts. An additional ship, Ship B, is selected that is not under way; this ship has a reporting precision of 1 count.

The length of time from a measurement to an estimate is variable. For example, if a ship reports every six seconds, the error will nominally be based on a six second prediction. In practice, the data set provided is missing some reports. This is particularly true when a ship is far away from the receiving station. An additional criterion in ship selection for this example is contiguous measurements, or a small number of missing measurements. The time between measurements for the first 2000 measurements of ships A, B and C is shown in Figure 4.2. Although there are instances of up to 90 seconds without a report, the plots are scaled such that time differences of greater than 40 seconds are not shown to illustrate the detail for the nominal differences. Notice the dip in time difference for Ship A around measurements 400 and 750. This represents a time when the ship is changing course, and the reporting rate increased accordingly. Ship A and Ship C are moving, and have a nominal reporting interval of six seconds. Ship B is nominally stationary, and has a nominal reporting interval of ten seconds. The following conclusions can be drawn from Figure 4.2: there is significant variation around the nominal reporting interval, and the data set is likely missing some reports.

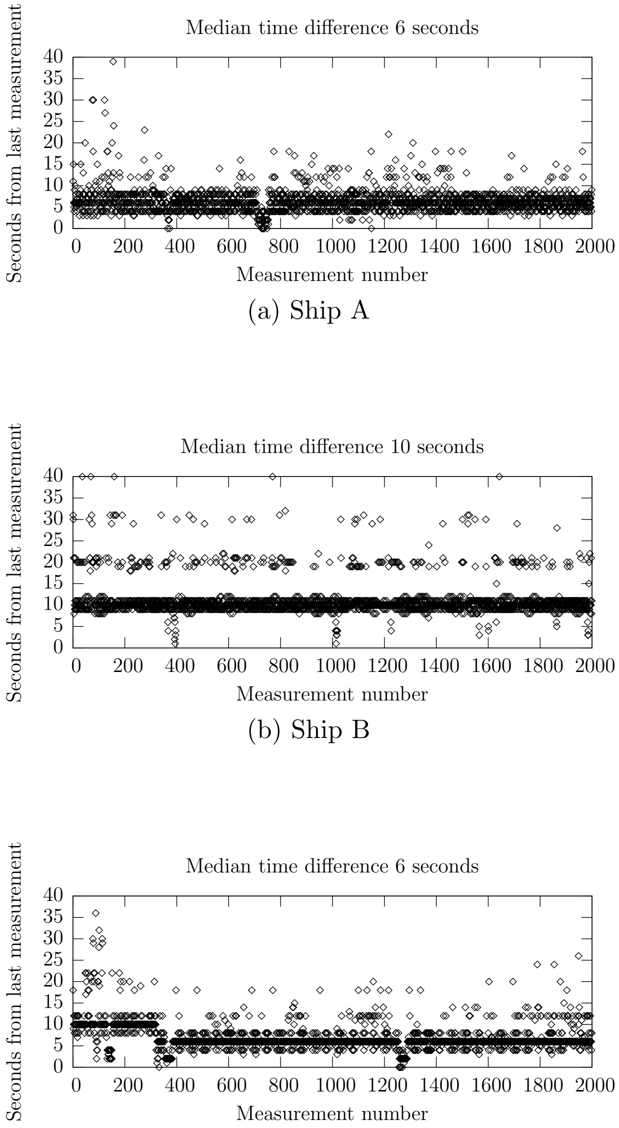
Let the error, $e[k]$, be defined as the difference between the reported (quantized) value, $q(\boldsymbol{x}_k)$ at time k and the estimation of the reported value, $q(\hat{\boldsymbol{x}}_{k|k-1})$, at time k based on all measurements before time k:

$$e[k] = q(\hat{\boldsymbol{x}}_{k|k-1}) - q(\boldsymbol{x}_k) \tag{4.12}$$

The $q(\cdot)$ operator represents the quantization operation resulting in the reported quantized value rather than the actual unquantized value.

The entropy (in bits) of the error term, e, with discrete probability distribution $(p_1, p_2, \ldots, p_K)$ is defined as

$$H(\boldsymbol{p}) = \sum_{k=1}^{K} p_k \log_2(1/p_k) \tag{4.13}$$



(a) Ship A

(b) Ship B

(c) Ship C

Figure 4.2: Time Between Position Reports

where we define $0 \log_2(1/0) = 0$. The distribution of the error term can be estimated with a normalized histogram.

1. Longitude Field

The latitude and longitude fields are unique in that there are many bits used to represent the value, yet there is little information contained in additional measurements after one measurement has been received. Prediction results for the longitude field are presented below. The results for the latitude field are similar.

A two-state Kalman filter is used to predict the longitude field. The state transition matrix is identical to the example of predicting the location of a train on a straight track discussed in Section IV.C

$$\boldsymbol{\Phi}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{4.14}$$

where the time step Δt is taken to be one second.

The process noise must be modeled large enough to model maneuvers that a ship may perform; this is effectively modeled as noise in the velocity term. Consider an unknown acceleration term, a, modeled as a Gaussian random variable. The $\boldsymbol{w}_k$ term in Equation (4.3) becomes

$$\boldsymbol{w}_k = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a_k \tag{4.15}$$

The random acceleration, a_k, is modeled as a zero-mean Gaussian random variable with standard deviation σ_{a_k}. For this experiment σ_{a_k} is chosen to be 0.71 counts/sec^2, where the units are binary counts, i.e. the raw values that are transmitted. There is no need to convert to physical units.

The covariance matrix for the process noise $\boldsymbol{Q}_k$ is then

$$\boldsymbol{Q}_k = 0.5 \begin{bmatrix} 1/4\Delta t^4 & 1/2\Delta t^3 \\ 1/2\Delta t^3 & \Delta t^2 \end{bmatrix}. \tag{4.16}$$

The variance of the measurement error is set to

$$\boldsymbol{R}_k = 20, \tag{4.17}$$

with units of counts squared. This value is determined empirically by looking at many sets of data. The variance of the measurement error represents both errors from the ship estimate of its position and quantization errors.

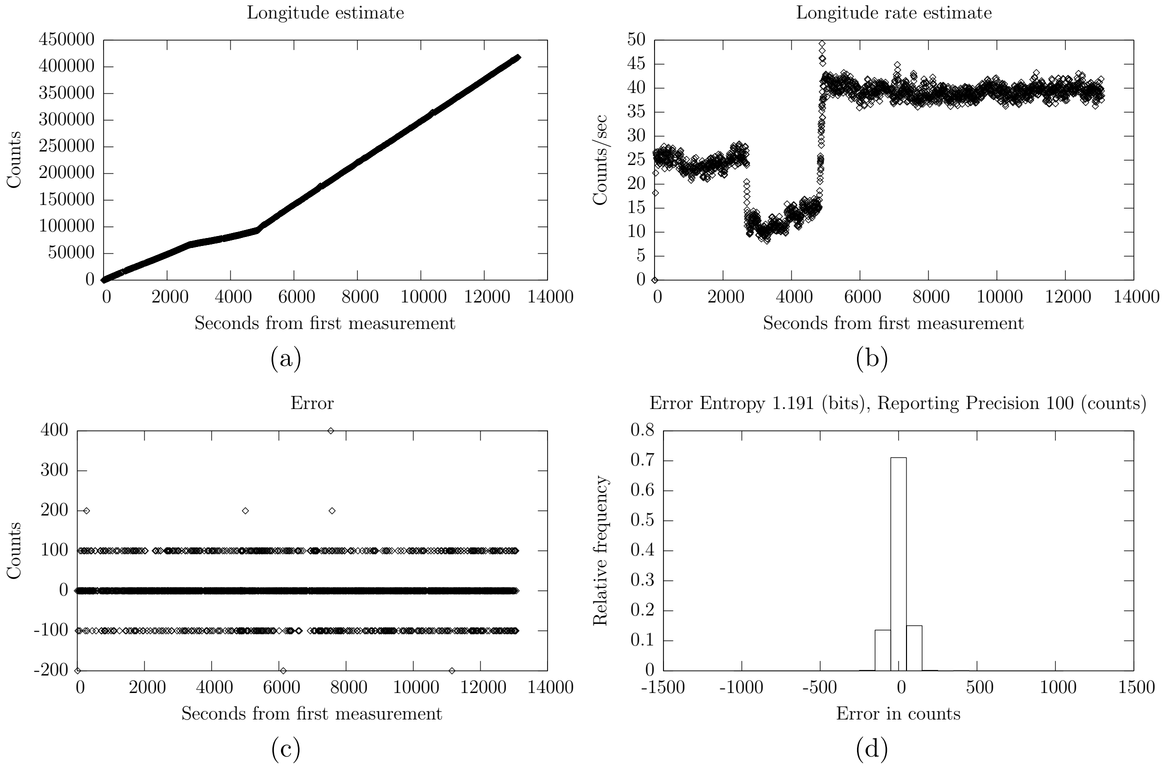
The initial estimate $\hat{\boldsymbol{x}}_0$ has the first element set to the initial measurement with the second element, representing velocity, set to zero. The initial error covariance is set to

$$\boldsymbol{P}_0 = \begin{bmatrix} 100 & 0 \\ 0 & 10^{10} \end{bmatrix}, \tag{4.18}$$

representing a large uncertainty for the initial estimate of the velocity term, and a relatively small uncertainty for the initial estimate of the position term.

The prediction results for the longitude field are presented for each of the three ships in Figures 4.3, 4.4, and 4.5. For each of the ships, the estimates of longitude and longitude rate are plotted. The error in the longitude estimate is also plotted versus time. A histogram of this error with a measure of the entropy of the error is also presented.

Figure 4.3 shows the prediction performance of the Kalman filter based predictor for a ship traveling at a nominally constant rate with a reporting precision of 100 counts. Figure 4.3b illustrates two clear changes in longitude rate. Notice in Figure 4.3c and Figure 4.3d that the distribution of values around the estimate is effectively just three values, the nominal estimate and one on each side. Also notice that the rate changes that occurs at approximately 2200 and 4600 seconds from the first measurement do not cause an increase in error in the estimate. The ability to track rate changes is incorporated into the process noise matrix $\boldsymbol{Q}_k$. The entropy of the error in the estimate, or the added information from an additional measurement is approximately 1.2 bits.



(a)

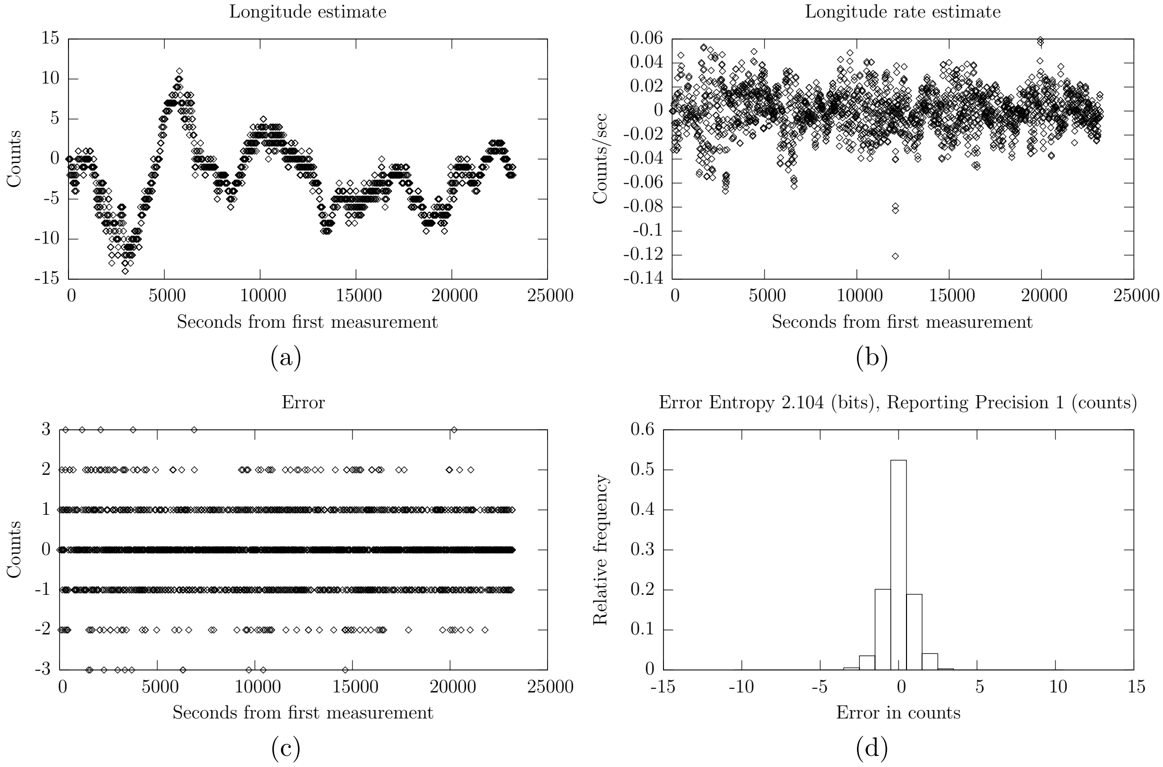
(b)

(c)

(d)

Figure 4.3: Longitude Field Prediction Performance Ship A

Figure 4.4 shows the results from a ship that is not moving that reported with a precision of 1 count. As indicated in Figure 4.4b, the velocity of Ship B is effectively zero for this data set. Notice in Figure 4.4d that the distribution of values around the estimate is effectively five values; two on each side of the estimate. The entropy of the error of the estimate, or the added information from an additional measurement, is approximately 2.1 bits. Although the error in the estimate is lower for Ship B than for Ship A, because of the higher reporting precision, the entropy of the error is actually higher.



(a)

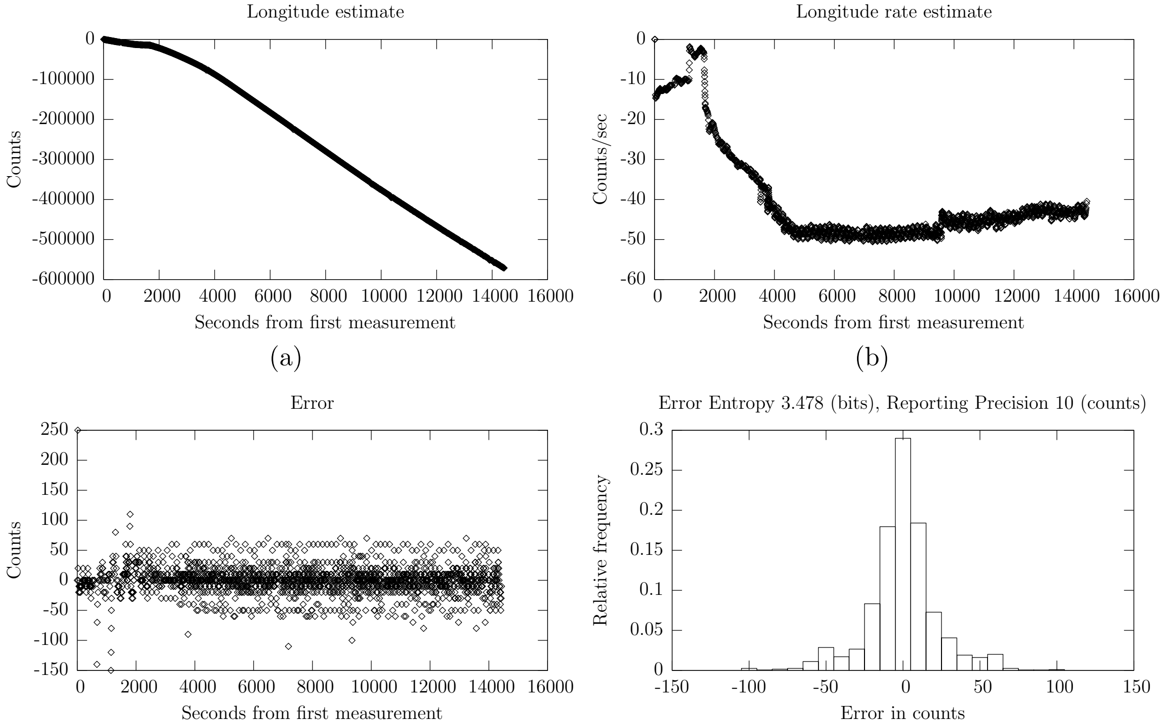
(b)

(c)

(d)

Figure 4.4: Longitude Field Prediction Performance Ship B

Figure 4.5 shows the results of a ship with similar motion to that of Ship A but with a reporting precision of 10 counts. In this example the error in the estimate increases as the magnitude of the rate of change in the longitude value increases. Changes in the velocity term of the Kalman filter are modeled with a noise term. It will take some time for the Kalman filter to settle on a new estimate for velocity; during this time the error in the estimate will be of a larger magnitude. There is some unexplained non-uniformity in the distribution of the error seen as some banding, or a set of outliers, in the error plot that has not been accounted for. This non-uniformity can also be seen in the histogram at around +50 counts and −50 counts as two small humps. Notice in Figure 4.5d that the distribution of values around the estimate is effectively 13 values, six on each side of the estimate. For this case the entropy of the error estimate, or the added information from an additional measurement is approximately 3.5 bits.



(a) (b)

(c) (d)

Figure 4.5: Longitude Field Prediction Performance Ship C

Some observations follow: Both the reporting precision and whether or not a ship is moving have an effect on the entropy of the AIS fields. It is apparent from the data from Ship C that the error can not necessarily be modeled as a stationary random process. The distributions for the longitude field will theoretically contain 2^{28} elements, but for all practical purposes only a small number of elements are required to represent the distribution when prediction strategies such as those presented here are used. The significance of this will become more apparent in Chapter VI which discusses the complexity of a MAP detector.

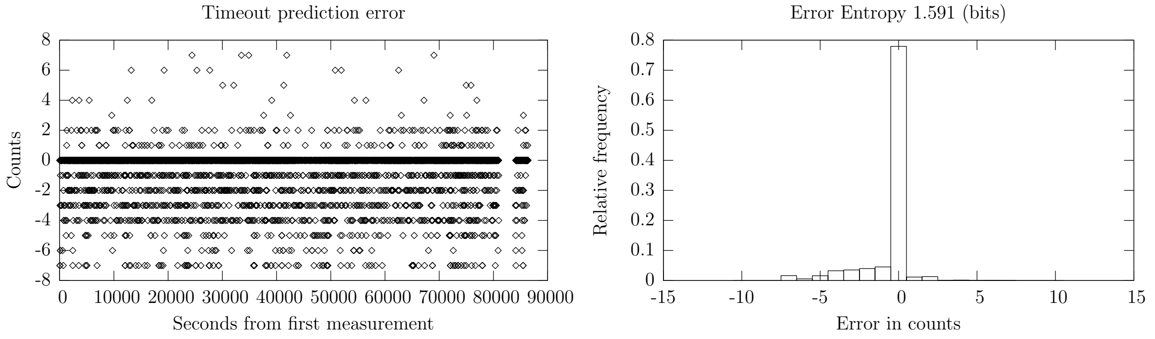
2. SOTDMA Communication State Field

The SOTDMA field is predicted using the algorithm summarized in Table 4.4. The residual entropy in the error from prediction of the Communication State field is not dependent on the reporting precision or the reporting rate. Therefore, only the results for one of the ships (Ship B) are presented.

The results from the Sync State field are trivial. This field did not change for any of the three ships for the duration of the data sets. The calculated entropy for this field would therefore be zero.

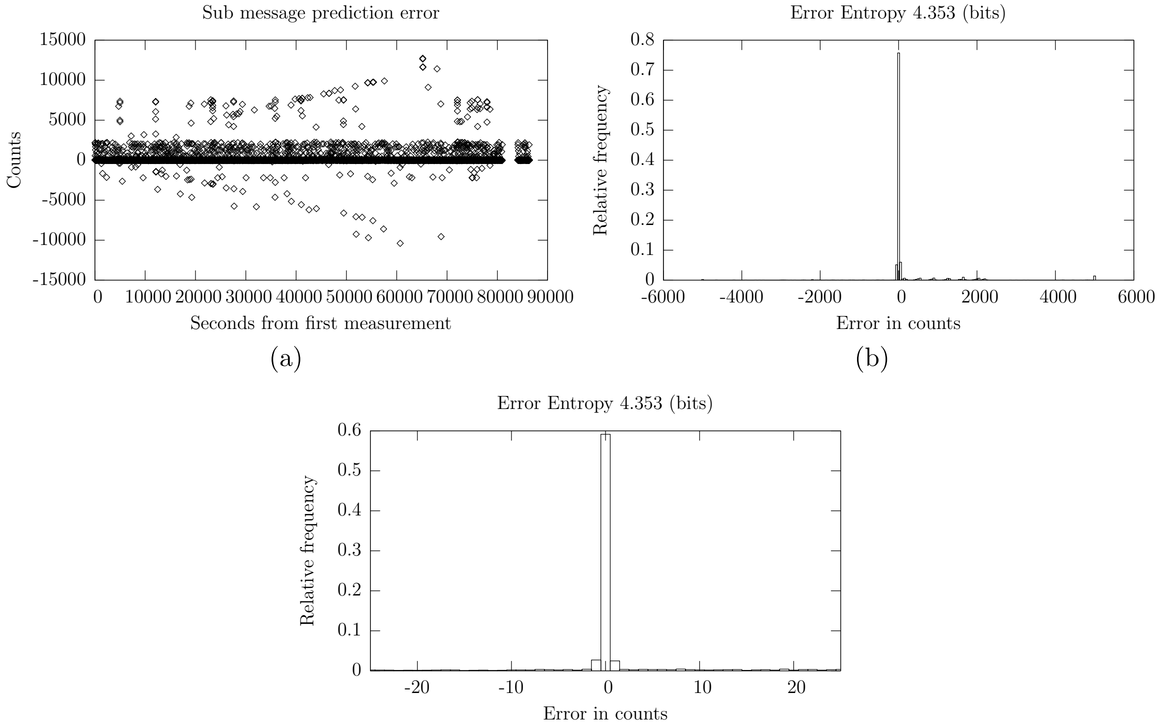
The prediction error for the Timeout field is presented in Figure 4.6a with a histogram showing the distribution of the error in Figure 4.6b. The Timeout field was correctly predicted approximately 79% of the time. For this data set the timeout was zero approximately 18% of the time. When the timeout is zero, the next value of the timeout is randomly selected from the set $\{1, 2, 3, \ldots, 7\}$. Therefore the probability of an error in prediction is 6/7. There are two conditions under which timeout is incorrectly estimated: the slot timed out and a new timeout is randomly selected, or a message is missed such that the previous timeout is unknown. When the slot times out, a timeout value of seven is arbitrarily selected. For this reason, most of the errors in the estimate are on one side of the distribution.

The results for the communication state submessage is shown in Figure 4.7. Close examination of Figure 4.7a reveals three types of errors corresponding to errors estimating the time of day, the slot, or the number of received stations. If the timeout value is incorrectly determined, that incorrect estimate will be used for determining which of the three potential values of the submessage to estimate, leading to three distinct errors. Observe from Figure 4.7b and 4.7c that the distribution of values around the estimate is tight for approximately 75% of the values with the remaining 25% of the values are spread over a large number of count values. There is a somewhat uniform distribution of errors among the 2250 possible slot values, another Gaussian type distribution for the number of receiving stations and then a time-varying distribution around the hour and minutes of the Time field. The calculated entropy for the submessage is approximately 4.4 bits, significantly less than the 14 bits that are transmitted for this field.



(a) (b)

Figure 4.6: Communications State Timeout Field Prediction Performance Ship B



(a) (b)

(c)

Figure 4.7: Communications State Submessage Field Prediction Performance Ship B

F. CHAPTER SUMMARY

This chapter develops an efficient method to predict the AIS data fields using three different types of predictors. An algorithm is developed for prediction of the AIS Communication State field. The results of these predictors are then presented on three representative data sets. The next Chapter discusses methods of incorporating knowledge of bit-stuffing into a MLSD. In the following Chapter Maximum *a posteriori* detection is discussed; here, the importance of distribution knowledge on performance and the importance of distribution reduction on computation time is illustrated.

V. MLSD FOR BIT-STUFFING

This chapter investigates maximum likelihood sequence detection (MLSD) for a bit-stuffed data source. The AIS uses the HDLC protocol which has bit-stuffing as one of its components.

Bit-stuffing is used in various communications protocols, typically as a way to remove ambiguity in the detection of control flags or as a method to aid clock recovery. Typically a demodulator makes hard symbol-by-symbol decisions without incorporating the knowledge that the data source is bit-stuffed. A demodulator making symbol-by-symbol decisions is clearly not optimum for a source that has been bit-stuffed; symbol-by-symbol decisions ignore the knowledge that the source has been encoded. In this chapter an optimum approach is investigated. The maximum likelihood sequence detector for a bit-stuffed data source is developed; starting by modeling the transmitted signal as a Markov chain. Approximate performance bounds for the MLSD are calculated. Finally, a trellis receiver structure is developed, and perform simulations using the Viterbi decoder algorithm are performed.

A. INTRODUCTION AND MODEL

Bit-stuffing is the procedure of introducing an extra bit after a fixed number of consecutive ones or zeros [93]. This chapter is concerned with the insertion of a zero after k consecutive ones; let this be indicated as 1^k bit-stuffing. The High-level Data Link Control (HDLC) protocol is an application where 1^5 bit-stuffing is used. Bit-stuffing has two obvious benefits: it aids in clock recovery, and it removes any ambiguity in locating the start and stop flags, 0x7E (01^60), for HDLC because these flags are not bit-stuffed.

Typically unstuffing is handled after demodulation. The demodulator is designed as if the data bits were independent, making symbol-by-symbol decisions. A demodulator making symbol-by-symbol decisions is clearly not optimum for a bit-stuffed source.

An alternate approach in demodulation is to take advantage of the knowledge available for the signal. For example, in 1^5 bit-stuffing the sequence 111111 is not a valid sequence; if a demodulator outputs that sequence it would be known that an error occurred. A maximum likelihood sequence detector will appropriately take advantage of this information; this detector is efficiently implemented using the Viterbi algorithm.

It should be noted that once the demodulator has made hard decisions nothing can be done to improve the error performance. For example, with 1^5 stuffing if six ones are received in a row, there has been an error, but the error could be in any of the six bits. There is a 1/6 probability of correcting the error, but there is a 5/6 probability of creating another error. Another approach is to use soft outputs to help decide which of the six possible bits were in error; this technique is not optimal, and will not be discussed here.

Bit-stuffing is similar to run-length-limited (RLL) codes. A (d, k) run-length-limited code has the property that consecutive 1's are separated by at least d but no more than k zeros. Bit-stuffed data has the property that consecutive zeros are separated by at least zero ones, and no more than k ones. Many authors have investigated error control coding for bit-stuffed data or RLL codes [94, 95]. We [96] are not aware of any other attempts to examine the performance of a maximum likelihood sequence detector for bit-stuffed data.

1. Bit-Stuffing

Let $\boldsymbol{a}$ be a vector of information bits. A new stuffed vector $\boldsymbol{a}_s$ is created by bit-stuffing the original vector $\boldsymbol{a}$. For this dissertation, bit-stuffing consists of the insertion of a zero after each consecutive string of k ones, where the leftmost bit is transmitted first. Let B_k denote the bit-stuff operator. For example, if

$$\boldsymbol{a} = 011101, \tag{5.1}$$

then

$$\boldsymbol{a}_s = B_2(\boldsymbol{a}) = 011\bar{0}101, \tag{5.2}$$

where the overbar indicates an inserted bit. Let B_k^{-1} denote the bit unstuffing operator such that

$$\boldsymbol{a} = B_k^{-1}(\boldsymbol{a}_s). \tag{5.3}$$

The received vector of decision statistics can be defined as

$$\boldsymbol{r} = \boldsymbol{a}_s + \boldsymbol{w}, \tag{5.4}$$

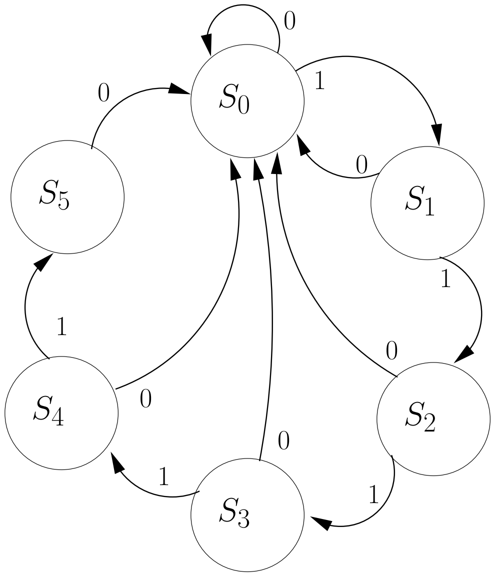


Figure 5.1: Model for 1^5 Markov Process Data Source

where $\boldsymbol{w}$ is a vector of samples from a bandlimited white noise process. The samples in $\boldsymbol{w}$ have variance σ^2.

2. Modeling the Bit-Stuffing Process

The sequence $\boldsymbol{a}_s$ can be modeled as a Markov chain. A Markov model for $\boldsymbol{a}_s = B_5(\boldsymbol{a})$, where $\boldsymbol{a}$ contains independent equally likely symbols, drawn from $\{0, 1\}$, is shown in Figure 5.1. Note that for each state with the exception of state S_5 the probability of a one is equal to the probability of a zero, $\Pr(0) = \Pr(1) = 0.5$. For state S_5, a zero is transmitted with probability one. It is this extra piece of information that a demodulator can exploit to obtain a slightly lower bit error rate (BER) than that of the symbol-by-symbol detector.

3. Demodulators

The first demodulator examined is a symbol-by-symbol maximum likelihood detector. This detector selects $\hat{a}_s[k]$ based on the following criteria

$$\hat{a}_s[k] = \arg \max_{a_s \in \{0,1\}} \Pr(r[k]|a_s[k]). \tag{5.5}$$

It should be clear that this first detector is not able to incorporate knowledge of valid or invalid sequences; decisions are based only on the current sample.

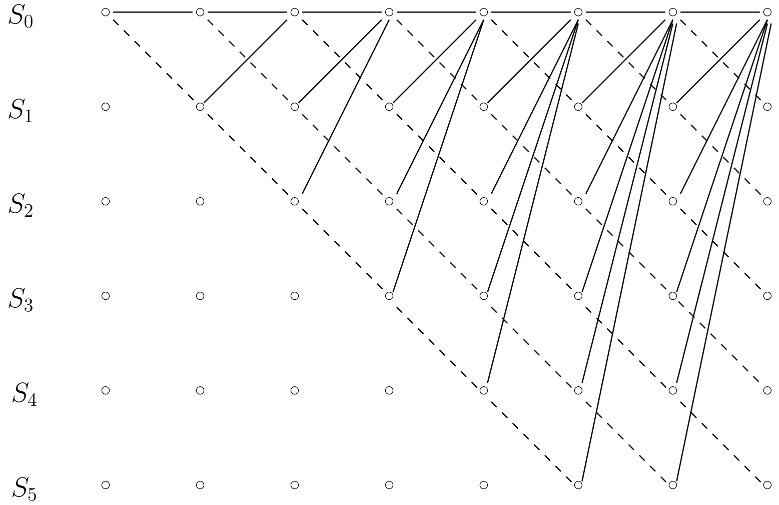


Figure 5.2: Trellis Structure for 1^5 Bit-Stuffed Markov Process

The second demodulator considered is the maximum likelihood sequence detector. This detector selects the entire vector $\hat{\boldsymbol{a}}_s$ based on the following

$$\hat{\boldsymbol{a}}_s = \arg \max_{\boldsymbol{a}_s \in A_s} \Pr(\boldsymbol{r}|\boldsymbol{a}_s) \tag{5.6}$$

where A_s is the set of valid stuffed sequences. The MLSD incorporates what is known about the data source into the demodulator. The MLSD is efficiently implemented using the Viterbi algorithm. An example of a trellis for decoding 1^5 bit-stuffed data is shown in Figure 5.2, where the process is in state S_n if the previous n bits were ones. Transitions due to a "1" are indicated with a dashed line, and transitions due to a "0" are indicated with a solid line. The only state with more than one branch entering is state S_0. The add-compare-select function in the Viterbi algorithm needs only to be applied at this state, reducing the computational complexity of the algorithm.

B. CATASTROPHIC ERRORS

Bit-stuffing is a catastrophic encoding process; a finite number of errors in the encoded stream $\boldsymbol{a}_s$ can lead to an infinite number of errors in the recovered data stream $\boldsymbol{a}$. Most applications that use bit-stuffing are packet-based so an infinite number of errors is reduced to a single packet error. An interesting question is "What is the probability of a catastrophic error given a single error in the encoded stream $\boldsymbol{a}_s$?" A catastrophic error occurs in the unstuffing operation when a bit is either removed erroneously or not removed when it should be.

There are two first order phenomena where a single flipped bit leads to a catastrophic error. First consider a bit removed erroneously due to a single error. This requires a special sequence. For example, consider 1^5 bit stuffing and the sequence below:

$$011\bar{0}11010110\bar{0}1111\bar{0}011111010$$

Any error in the bits with the overbar will cause a bit to be erroneously removed leading to a catastrophic error in the unstuffed output. Let these bits with the overbar be called type 1 bits. A type 1 bit for 1^k stuffing is a zero that can be flipped to create a non-overlapping sequence of k sequential ones.

Next, consider a stuffed bit erroneously not removed from the same sequence.

$$0110110101100111100\overline{11111}010.$$

Errors in the bits with the overbar will cause a stuffed bit to be erroneously left in the stream leading to a catastrophic error in the unstuffed output. Let these be called type 2 bits. A type 2 bit for 1^k stuffing is any of the k ones in a sequence of k sequential ones.

The probability (P_{cbe}) of a catastrophic error beginning at a particular bit is approximated by

$$P_{cbe} \approx P_{sbe}P_{t1} + P_{sbe}P_{t2}, \tag{5.7}$$

where P_{t1} and P_{t2} are the probabilities that a particular bit is a type 1 or type 2 bit, and P_{sbe} is the probability of any particular stuffed bit is received in error. This is an approximation and not an equality because higher order error events have been ignored, such as two errors leading to the erroneous removal of a bit, or the case of a second error canceling the effects of the first.

The probability of a frame containing a catastrophic error is

$$P_{cfe} = 1 - (1 - P_{cbe})^n, \tag{5.8}$$

where n is the length of the frame.

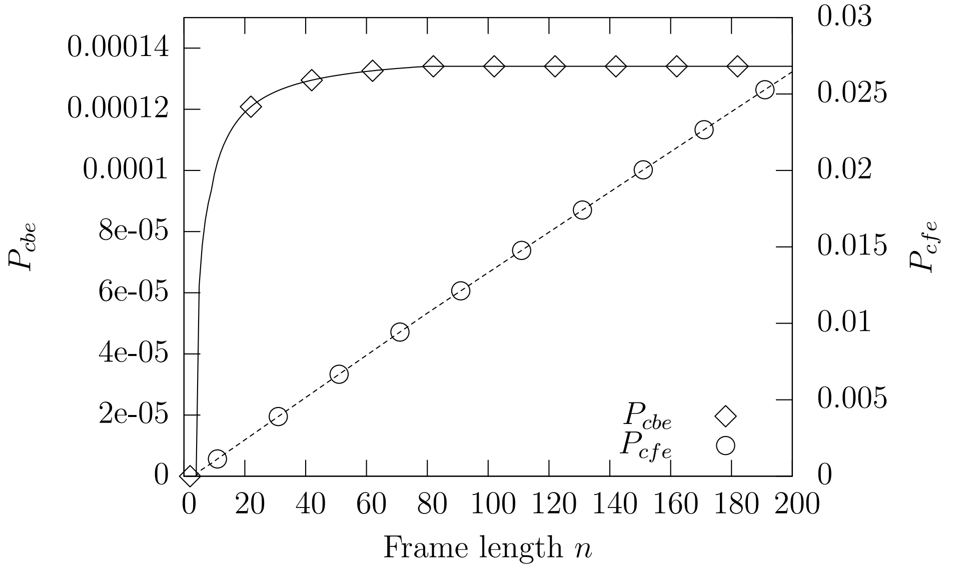


Figure 5.3: P_{cbe} and P_{cfe} versus Frame Length n, 1^5 Stuffing and $P_{sbe} = 10^{-4}$

A plot of the probability that a catastrophic error will begin at a certain bit, P_{cbe}, along with the probability of a catastrophic error occurring during a frame, P_{cfe}, are shown in Figure 5.3. Notice that P_{cbe} grows rapidly with frame length and then quickly levels out to be approximately constant. For small probability of stuffed bit error and frame lengths greater than around 20, P_{cfe} will increase linearly until the probability of multiple errors becomes significant at which point it will slowly flatten out and approach one.

C. CALCULATION OF P_{t2}

There has been some interest in computing P_{t2} by those interested in calculating the probability that there will be N or more stuffed bits in a given sequence. Nolte, et al., [97] performed brute force calculations of the number of non-overlapping runs of k ones in n bits for their work in latencies in the Controller Area Network (CAN) bus. An alternative approach is demonstrated below.

Let $N_{n,k}$ be the number of non-overlapping runs of k ones in n bits. Begin by calculating the probability of one or more non-overlapping runs of k ones in a stream of n random bits $\Pr(N_{n,k} \geq 1)$.

Markov chains can be used for this calculation without resorting to brute force or Monte Carlo approaches [98]. Consider a sequence of n independent bits where $\Pr(1) = \rho$,

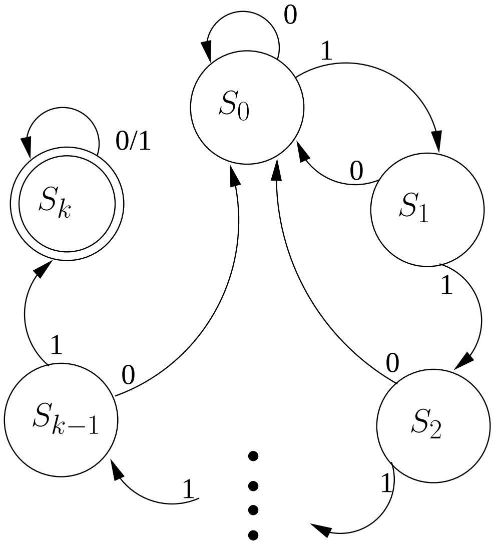


Figure 5.4: Markov Chain for Calculating $\Pr(N_{n,k} \geq 1)$

and $\Pr(0) = 1 - \rho$. A state diagram for the Markov chain used for finding $\Pr(N_{n,k} \geq 1)$ is shown in Figure 5.4. The states are defined such that the Markov chain is in state S_k when $N_{n,k} \geq 1$ and in state S_i, for $0 \leq i < k$, when $N_{n,k} = 0$ and the last i bits are ones. Except for state S_k, each of the states advances with probability ρ or returns to state S_0 with probability $1 - \rho$. State S_k is referred to as the absorbing state; once entered it is never exited.

The $(k+1) \times (k+1)$ state transition matrix for the Markov chain is

$$\mathbf{\Pi_1} = \begin{bmatrix} 1-\rho & 1-\rho & 1-\rho & \dots & 1-\rho & 0 \\ \rho & 0 & 0 & \dots & 0 & 0 \\ 0 & \rho & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho & 1 \end{bmatrix}, \tag{5.9}$$

where $\Pi_1[i,j] = \Pr(S_{t+1} = i | S_t = j)$. The initial state probability vector is $\boldsymbol{\xi}_0 = \boldsymbol{e}_0$, where $\boldsymbol{e}_i$ is a length $(k+1)$ unit column vector with a 1 in the $(i+1)$-th position and zeros elsewhere, $\boldsymbol{e}_0 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0. \end{bmatrix}^T$. The state probability vector after n bits is given by

$$\boldsymbol{\xi}_n = \mathbf{\Pi_1}^n \boldsymbol{e}_0, \tag{5.10}$$

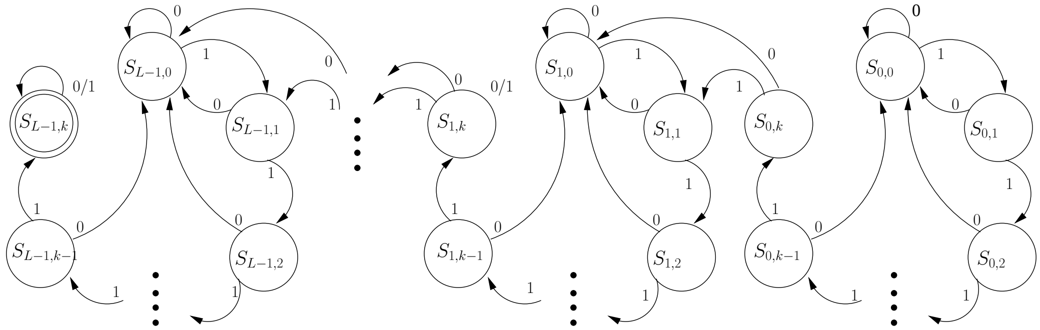


Figure 5.5: Markov Chain for Calculating $\Pr(N_{n,k} \geq L)$

where $\boldsymbol{\xi}_n$ is the state probability vector at time n. The probability of at least one sequence of k ones in n bits is the probability of being in state k, which can be written as

$$\Pr(N_{n,k} \geq 1) = \boldsymbol{e}_k^T \boldsymbol{\Pi_1}{}^n \boldsymbol{e}_0. \tag{5.11}$$

The probability that there are no sequences of k ones in n bits is given by

$$\Pr(N_{n,k} = 0) = 1 - \Pr(N_{n,k} \geq 1). \tag{5.12}$$

The above can be extended to determine $\Pr(N_{n,k} \geq L)$, the probability of L sequences of k ones in a stream of n random bits. A state diagram for the Markov chain used for finding $\Pr(N_{n,k} \geq L)$ is shown in Figure 5.5. The extended Markov chain consists of L cascaded rings, $L-1$ rings of identical form ending with a ring of the form identical to Figure 5.4. The states are defined such that the Markov chain is in state $S_{L-1,k}$ if $N_{n,k} \geq L$, in state $S_{i,k}$ for $0 \leq i < L-1$ if $N_{n,k} = i+1$ and the most recent k bits are ones, and in state $S_{i,j}$ for $0 \leq i < L-1$ and $0 \leq j < k$ if $N_{n,k} = i$ and the last j bits are ones.

The $L(k+1) \times L(k+1)$ transition matrix for the extended Markov chain is

$$\boldsymbol{\Pi_L} = \begin{bmatrix} \boldsymbol{\Pi_1}' & \boldsymbol{Z} & \dots & \dots & \boldsymbol{Z} \\ \boldsymbol{N} & \boldsymbol{\Pi_1}' & \dots & \dots & \boldsymbol{Z} \\ \boldsymbol{Z} & \boldsymbol{N} & \ddots & \dots & \boldsymbol{Z} \\ \vdots & \vdots & \dots & \boldsymbol{\Pi_1}' & \boldsymbol{Z} \\ \boldsymbol{Z} & ... & \dots & \boldsymbol{N} & \boldsymbol{\Pi_1} \end{bmatrix} \tag{5.13}$$

where $\Pi_L[j+(k+1)i, m+(k+1)l] = \Pr\{S(t+1) = S_{i,j} | S(t) = S_{l,m})\}$, $\boldsymbol{Z}$ is a $(k+1)\times(k+1)$ zero matrix and $\boldsymbol{N}$ is a $(k+1) \times (k+1)$ zero matrix with the exception of two terms in the upper right corner representing the transition probabilities to the $S_{l,0}$ and $S_{l,1}$ states.

$$\boldsymbol{N} = \begin{bmatrix} 0 & \dots & 0 & 1-\rho \\ 0 & \dots & 0 & \rho \\ \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.14}$$

$\boldsymbol{\Pi_1}'$ is identical to the $\boldsymbol{\Pi_1}$ matrix without the absorbing state $S_{l,N}$.

$$\boldsymbol{\Pi_1}' = \begin{bmatrix} 1-\rho & 1-\rho & 1-\rho & \dots & 1-\rho & 0 \\ \rho & 0 & 0 & \dots & 0 & 0 \\ 0 & \rho & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho & 0 \end{bmatrix} \tag{5.15}$$

The probability of being in a particular state after n bits is given by

$$\boldsymbol{\xi}_n = \boldsymbol{\Pi_L}^n \boldsymbol{e}_0, \tag{5.16}$$

and the probability of at least L sequences of k ones in n bits is then

$$\Pr(N_{n,k} \geq L) = \boldsymbol{e}_{L(k+1)-1}^T \boldsymbol{\Pi_L}^n \boldsymbol{e}_0, \tag{5.17}$$

as before.

Consider determining P_{t2}, P_{t2} is the probability that a specific bit in $\boldsymbol{a}_s$ is one of the particular bits that when flipped will cause a “type 2” catastrophic error. This is found by summing over the probability of exactly l sequences of length k, multiplying by the number of bits (k) in those sequences and dividing by the total number of bits (n). Let $\mathcal{B}$

describe the event that a bit is in a run k ones. P_{t2} is then

$$\begin{aligned} P_{t2} &= \sum_{l=1}^{\lfloor n/k \rfloor} \Pr(\mathcal{B}|N_{n,k} = l) \Pr(N_{n,k} = l) \\ &= \frac{k}{n} \sum_{l=1}^{\lfloor n/k \rfloor} \Pr(N_{n,k} = l) l. \end{aligned} \tag{5.18}$$

$\Pr(N_{n,k} = l)$ can be calculated using (5.17) as follows

$$\Pr(N_{n,k} = l) = \Pr(N_{n,k} \geq l) - \Pr(N_{n,k} \geq l + 1). \tag{5.19}$$

It should be noted that Philippou and Makri [99], and independently in the same year Hirano [100], developed a formula for calculating $\Pr(N_{n,k} = l)$. Unfortunately it is not clear how to extend their results to find P_{t1}.

D. CALCULATION OF P_{t1}

Consider the calculation of P_{t1}, P_{t1} is the probability that any bit in $\boldsymbol{a}_s$ is one of the particular zero bits that when flipped will create a non-overlapping sequence of k sequential ones leading to a “type 1” catastrophic error. For example, for 1^2 stuffing the sequence $\overline{0}1\overline{0}0\hat{0}11$ has type 1 bits located at the zeros with the overbar, but the zero with the hat is not a type 1 bit, from the non-overlapping requirement. Calculating the probabilities of these locations is unfortunately more difficult than the case of type 2 errors, but the same general technique that was used to determine P_{t2} can be applied. The sequences take the form $1^{j-1}01^{k-j}$, where j is the location of the zero in the sequence of ones, $j = 1$ being the leftmost bit position. Let $N_{n,k}^{1^{j-1}01^{k-j}}$ be the number of runs of $1^{j-1}01^{k-j}$ that do not overlap with a run of k ones. A Markov chain for finding $\Pr(N_{n,k}^{1^{j-1}01^{k-j}} \geq 1 \ \forall j \in \{1, 2, \ldots, k\})$ for 1^4 bit-stuffing is shown in Figure 5.6. The Markov chain can be extended as before to calculate the individual terms, and find

$$P_{t1} = \frac{1}{n} \sum_{l=1}^{2 \times \lceil n/k \rceil} \Pr(N_{n,k}^{1^{j-1}01^{k-j}} = l) l. \tag{5.20}$$

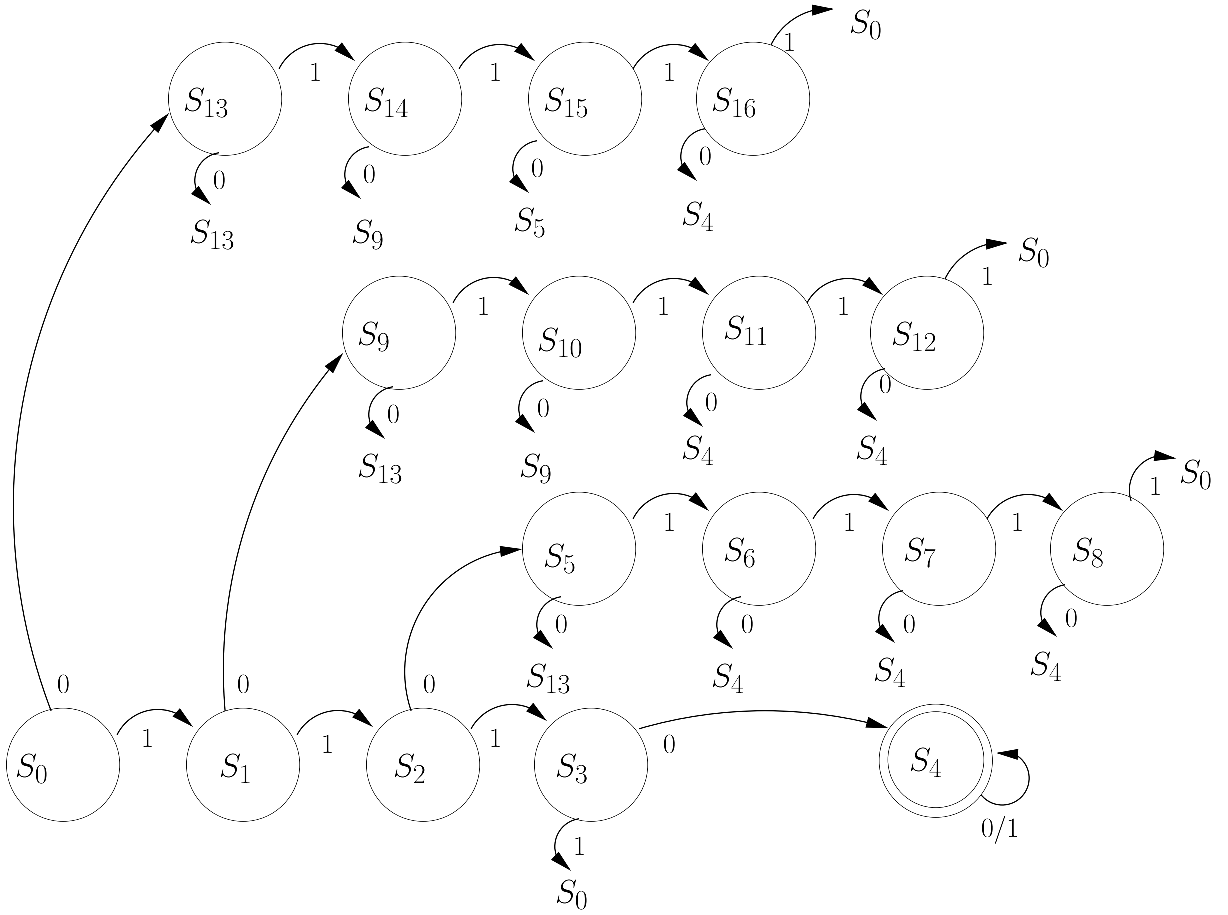


Figure 5.6: Markov Chain for Calculating $\Pr(N_{n,k}^{1^{j-1}01^{k-j}} \geq 1\ \forall j \in \{1, 2, \ldots, k\})$ for 1^4 Bit-Stuffing

Finally, Equation (5.7) may be used to find P_{cbe}. To understand the limits on the sum in Equation 5.20 consider the following sequence for B^2: "010101010" with $n = 7$ and $k = 2$. The number of type 1 bits in this sequence is four. The limit represents the maximum number of special bits that could occur in a sequence.

E. BOUNDS ON BIT ERROR RATE PERFORMANCE

Deriving an analytical expression for error performance is slightly more complicated than finding an error performance bound for a linear convolutional code. Assuming the all zeros sequence is sent results in an overly pessimistic error bound. This is because the invalid codes are relatively far in Euclidean distance from the all zero sequence. Forney [51] showed that a tight lower bound for probability of error for a MLSD is

$$P_{sbe} \geq K_0 Q(d_{min}/2\sigma), \tag{5.21}$$

where σ is the standard deviation of the noise component of the decision statistic. For this example, K_0 is the probability that an error of Euclidean distance d_{min} from the given input sequence will lead to a valid sequence. A lower bound on performance can be found relatively easily using the results from Section V.B.

The encoding format from bit-stuffing reduces the set of possible valid sequences. To calculate a lower bound, first find the sequences that have a distance of d_{min} to one of the invalid sequences. The lower bound on probability of bit error is then the average number of d_{min} error events divided by the number of bits. These events are similar to the "type 1" events, but not identical. For example, for 1^5 bit-stuffing the sequences that have distance of 1 from an invalid sequence are 111110, 111101, 111011, 110111, 101111 or 011111. The events of interest are the zeros in the 1^5 stream $\boldsymbol{a}_s$, where the number of consecutive ones to the left and the right sum to at least $k = 5$. Let $Z_{n,k}$ be the number of zeros in a sequence of n bits stuffed using B_k with k ones to the left and right of each zero. As before $\Pr(Z_{n,k} \geq l)$ is found using Markov chains. This involves a Markov chain very similar to the one used to find "type 1" errors. Figure 5.7 illustrates this for 1^3 stuffing. Note that while this is very similar the Markov model used in calculating "type 1" errors for 1^4 stuffing, it is distinct beyond the change in bit-stuffing level; there is no non-overlapping requirement. In addition, after k ones a zero is inserted, so the chain jumps to a different state. It is important to notice that the Markov chain is run on data prior to bit-stuffing ($\boldsymbol{a}$), but it is predicting values for the bit-stuffed data ($\boldsymbol{a}_s$). The sequence of bits where this comes into play is 1^k or 01^k. In both of these cases it is known from the bit-stuffing rule that the next bit in $\boldsymbol{a}_s$ will be a zero and it will be one of the special zeros. Similar to P_{t1} the results from the Markov chain are used to calculate K_0

$$K_0 = 1 - \frac{1}{n} \sum_{l=1}^{2\times\lceil n/k \rceil} \Pr(Z_{n,k} = l) l \tag{5.22}$$

Figure 5.8 shows calculated values of K_0 for a few values of k versus the number of bits n. Clearly smaller values of k and larger values of n will lead to a larger potential performance improvement for the MLSD over symbol-by-symbol detection.

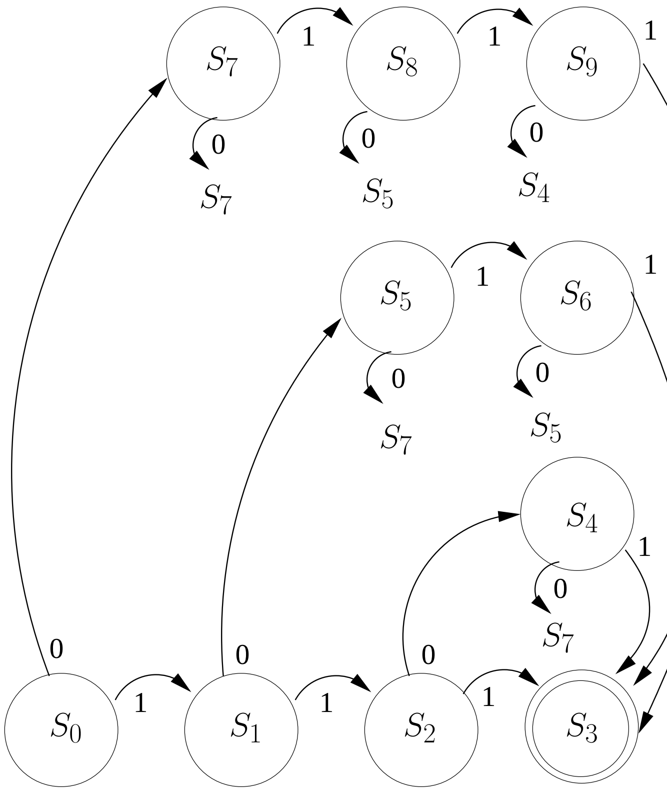


Figure 5.7: Markov Chain for Calculating K_0

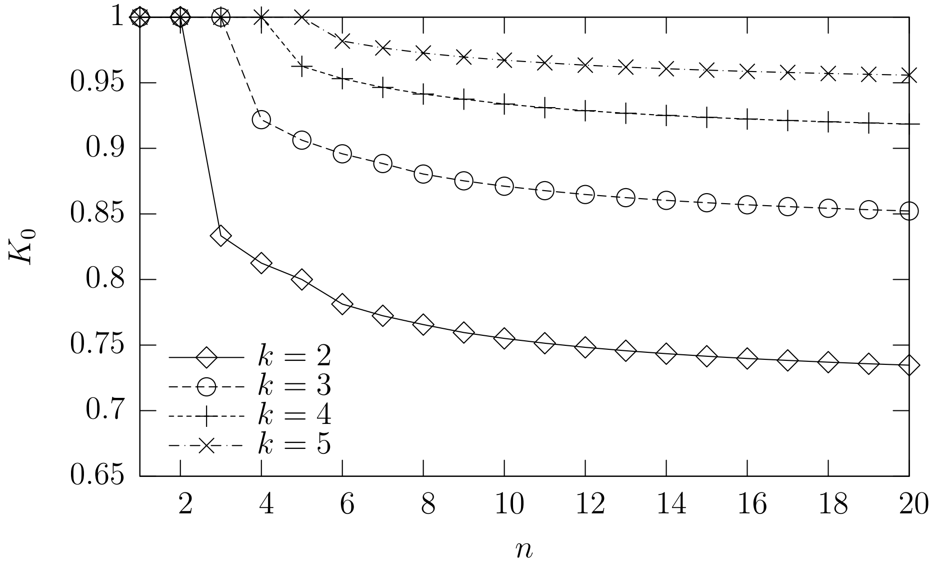


Figure 5.8: K_0 versus n

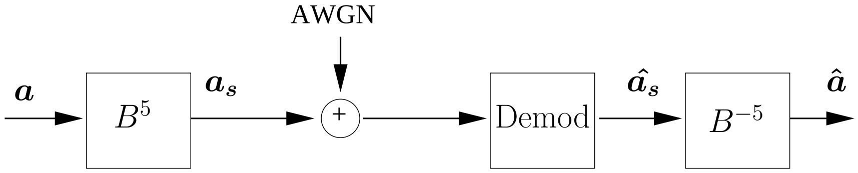


Figure 5.9: Illustration of Stuffing, Demodulation and Unstuffing Process

The lower bound (5.21) is tight for high SNR. For low SNR the neglected terms representing higher order events, more than one bit in error, become quite large and it is no longer a useful bound.

F. SIMULATION

1. Results

Data generated according to (5.4) is used as input to two demodulators: the symbol-by-symbol maximum likelihood detector and the maximum likelihood sequence detector described above. Each demodulator generates an estimate, $\hat{\boldsymbol{a}}_s$, of the vector $\boldsymbol{a}_s$. The vector $\hat{\boldsymbol{a}}$, an estimate of $\boldsymbol{a}$, is then generated using (5.3). See Figure 5.9 for an illustration.

Figure 5.10 shows the BER performance of the two demodulators for 1^5 bit-stuffed data with frame length $n = 168$ prior to unstuffing ($\boldsymbol{a}_s$) and after unstuffing ($\boldsymbol{a}$). Figure 5.11 shows the BER performance of the two demodulators for 1^2 stuffing. For 1^5 bit-stuffing and a frame length of 168 the term K_0 is approximately 0.9458. For frame lengths > 20, the BER of $\boldsymbol{a}_s$ using MLSD is approximately independent of frame length. Clearly from Figure 5.8 and (5.21) the lower bound on BER for $\boldsymbol{a}_s$ using MLSD is a function of frame length for small values of n.

For reasonably high SNR the BER performance for $\boldsymbol{a}_s$ using MLSD is close to the lower bound $K_0\,Q(\sqrt{2E_b/N_0})$. This represents a modest gain in BER for the stuffed vector $\boldsymbol{a}_s$.

Figure 5.12 shows the gain in performance as a reduction in required E_b/N_0 to achieve an equal BER. Gains using the MLSD are on the order of 0.05 dB for $\boldsymbol{a}_s$. For $\boldsymbol{a}$ the gains using the MLSD are from $0.2 - 1.0$ dB.

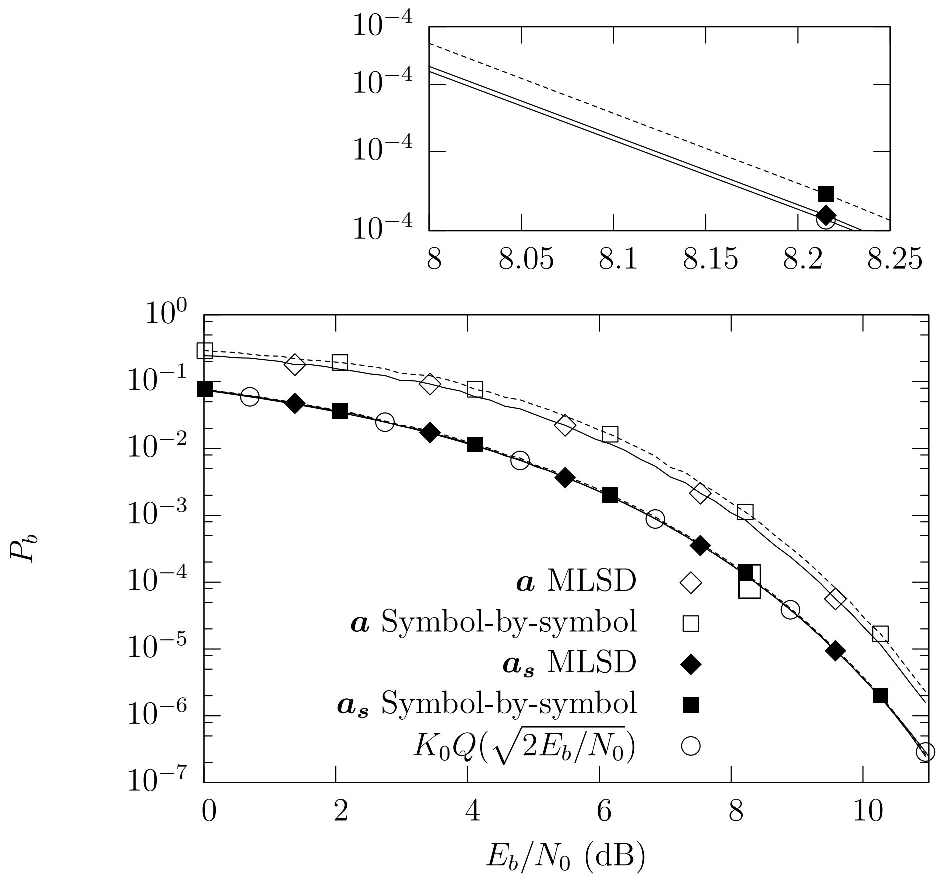


Figure 5.10: BER Curves Comparing MLSD and Symbol-by-Symbol Decisions for 1^5 Stuffing

2. Interpretation and Discussion of Results

The results from the previous section are BER curves for the stuffed vector $\boldsymbol{a}_s$. It is interesting to compare the performance of the detectors with the unstuffed vector $\boldsymbol{a}$. The number of errors caused by a catastrophic error is a function of frame length. A longer frame leads to a larger penalty for a catastrophic error. For this simulation a frame length of 168 bits is used.

The BER performance difference between the MLSD and the symbol-by-symbol detector is greater when measured for $\boldsymbol{a}$ then for $\boldsymbol{a}_s$. As described above, a single error in $\boldsymbol{a}_s$ can lead to multiple errors in $\boldsymbol{a}$. This is the reason for the higher BER for $\boldsymbol{a}$. The vector $\boldsymbol{a}$ contains the original information so it is generally the BER curve of $\boldsymbol{a}$ that is

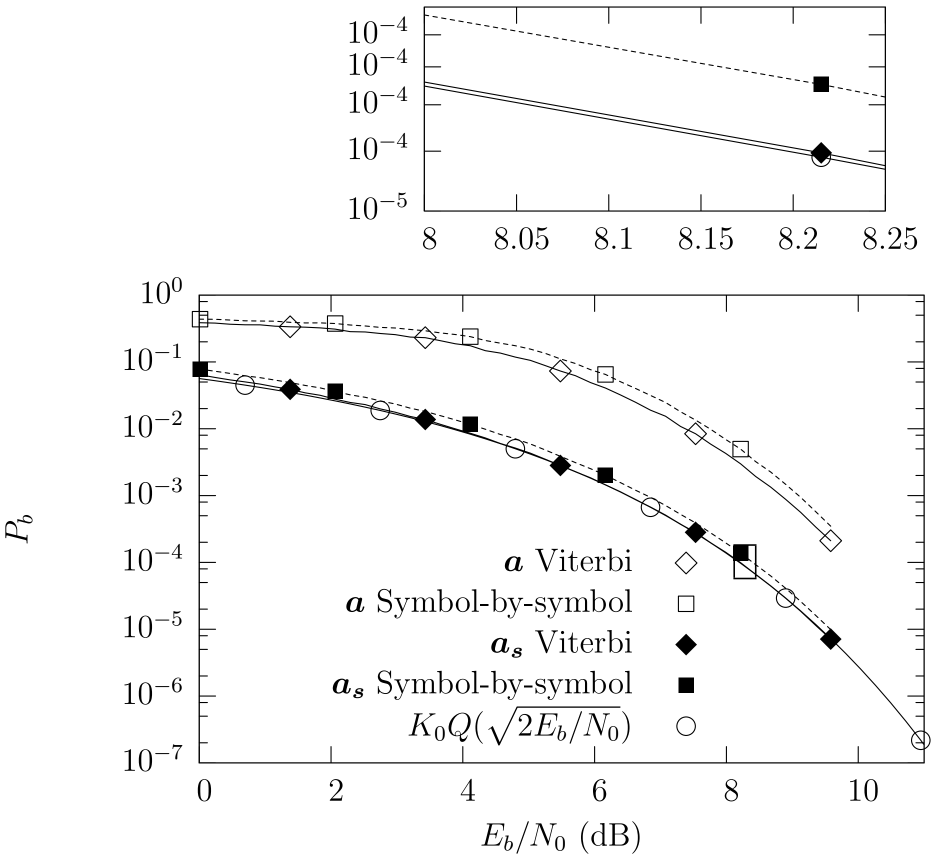


Figure 5.11: BER Curves Comparing MLSD and Symbol-by-Symbol Decisions for 1^2 Stuffing

important. The reason for the performance improvement is that the MLSD will decrease the likelihood of some of the decisions that would lead to catastrophic errors. Consider 1^5 stuffing and the sequence $\boldsymbol{a}_s = 1\bar{0}1111\bar{0}10$. Symbol-by-symbol decision errors at either of the two over bar zero positions would lead to a catastrophic error in $\boldsymbol{a}$. An error event of $2d_{min}$ would be required of the MLSD in order for a catastrophic error to take place.

G. CHAPTER SUMMARY

This chapter investigates MLSD for a bit-stuffed data source. The superior BER performance of MLSD versus symbol-by-symbol decisions is demonstrated, and a lower bound on performance is derived. The BER performance for both the stuffed stream $\boldsymbol{a}_s$ and the unstuffed stream $\boldsymbol{a}$ have been investigated demonstrating the effect of catastrophic

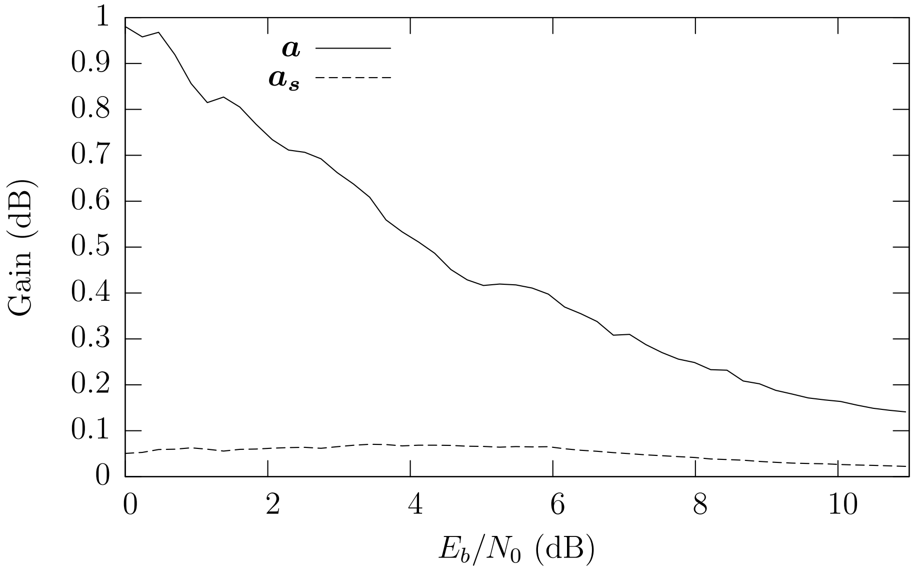


Figure 5.12: Gain (dB) Comparison of MLSD and Symbol-by-Symbol Decisions for 1^5 stuffing

errors. The probability of a catastrophic error for a given stuffed bit error rate P_{sbe} is derived. A bound for the BER performance of the stuffed stream $\boldsymbol{a}_s$ is developed by calculating probability of sequences with a Hamming distance of 1 from an invalid sequence.

For this analysis the case where the number of stuffed bits is known is not considered. It is possible that the number of stuffed bits for a particular frame would be known. For example, a protocol such as HDLC has a fixed unstuffed frame length while the stuffed frame length for HDLC can be calculated by counting the bits between the start and stop flags. The difference between these lengths will yield the number of stuffed bits. It is the author's belief that this additional knowledge can be used to further increase performance, particularly in the unstuffed vector $\boldsymbol{a}$.

The next Chapter investigates the multicategory classification problem of Field-Based MAP detection.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. FIELD-BASED MAP DETECTION

This chapter develops the MAP detector for an individual field of symbols. Here, a field is defined as a consecutive sequence of related symbols. A typical transmitted signal may contain many consecutive fields; here we consider the reception of a single isolated field through an AWGN channel with no ISI introduced at the transmitter. Later in the chapter the special case of reception of multiple *synchronous* signals is investigated. Both of these cases result in a multicategory classification problem for a multivariate Gaussian density function.

There are applications where data are transmitted as a sequence of fields, with each field containing a number of symbols. Here, the field values are independent, but the individual symbols contained in a field are not independent. For the example of AIS, the *a priori* knowledge available is for a field of symbols rather than priors on individual symbols. A detector that takes advantage of the additional information contained in the field *a priori* probabilities will lead to a lower probability of field error than that of either the MAP or ML symbol-by-symbol detector. Given that the fields are independent with no ISI, the optimum detector for a field of symbols can be developed by observing only the current field.

For this development the interest is in minimizing the probability of field error. While in general this will tend to also minimize probability of bit error, it will not always be the case, depending on the prior probabilities for particular field values. Field error is often a more natural measure of performance. If the field values contain the important information, one error or one hundred errors within a field result in the same cost.

A. SINGLE-CHANNEL FIELD-BASED MAP DETECTOR

Consider an information vector $\boldsymbol{a}$, where there is *a priori* knowledge of the relationship between a sequence of information symbols (a field). For example, we may know $\Pr(a_k, a_{k+1}, \ldots, a_{k+n} = \boldsymbol{b})$ where $\boldsymbol{b}$ is one particular realization of the n-tuple of all possible information symbols. The information vector $\boldsymbol{a}$ is mapped to a symbol vector $\boldsymbol{\alpha}$. This vector is received in the presence of AWGN resulting in the received vector

$$\boldsymbol{r} = \boldsymbol{\alpha} + \boldsymbol{z}, \tag{6.1}$$

where the elements of $\boldsymbol{z}$ are zero-mean independent identically distributed (i.i.d.) Gaussian random variables.

The maximum *a posteriori* (MAP) detector selects the field that maximizes $\Pr(\boldsymbol{\alpha}|\boldsymbol{y})$ [3, 101]

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}}[\Pr(\boldsymbol{\alpha}|\boldsymbol{y})]. \tag{6.2}$$

This decision rule results in the minimum probability of field error.

1. Model

This development of the multiple decision MAP field detector loosely follows examples in [101] and [102]. Consider the reception of a length N sequence of symbols in AWGN

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{z}, \tag{6.3}$$

where $\boldsymbol{\alpha}$ is a $N \times 1$ vector representing a field. Each element of $\boldsymbol{\alpha}$ is a member of $\{-A, A\}$. The noise vector $\boldsymbol{z}$ is an $N \times 1$ vector of zero-mean i.i.d. Gaussian random variables with variance σ^2. The vector $\boldsymbol{\alpha}$ represents one of the 2^N possible field values. Let only $M < 2^N$ field values have a non negligible probability of occurring. For each of the M likely field values, there is an associated prior probability of occurrence, $\Pr[\boldsymbol{\alpha}_k] = p_k \neq 0$.

Each element of the vector $\boldsymbol{z}$ has the same variance, and is independent of each of the other elements. This leads to the conditional PDF of $\boldsymbol{y}$ given by

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = (2\pi\sigma^2)^{-N/2} \exp\left(\frac{-(\boldsymbol{y}-\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{\alpha})}{2\sigma^2}\right). \tag{6.4}$$

The MAP detector selects the particular field $\hat{\boldsymbol{\alpha}} \in \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_M\}$ that maximizes $\Pr(\boldsymbol{\alpha}|\boldsymbol{y})$. Using Bayes' theorem and eliminating terms not dependent on $\boldsymbol{\alpha}$, it can be shown this is equivalent to selecting $\boldsymbol{\alpha}$ that maximizes $p(\boldsymbol{y}|\boldsymbol{\alpha})\Pr(\boldsymbol{\alpha})$. This leads to the following decision rule:

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}}[\Pr(\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{\alpha})]$$

Where $\mathcal{A}$ is the set $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_M\}$. From Equation (6.4) this leads to:

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \left[\Pr(\boldsymbol{\alpha}) \exp\left(\frac{-(\boldsymbol{y}-\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{\alpha})}{2\sigma^2} \right) \right]. \tag{6.5}$$

Any monotonic function of the right hand side of Equation (6.5) will lead to an identical result. The form of Equation (6.4) leads to the natural logarithm as a good choice, resulting in

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \left[\ln(\Pr[\boldsymbol{\alpha}]) - \left(\frac{(\boldsymbol{y}-\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{\alpha})}{2\sigma^2} \right) \right]. \tag{6.6}$$

2. Discriminant Functions

Another way to pose the same problem is in terms of discriminant functions. Discriminant function is a term borrowed from the pattern classification field. The discriminant function, denoted as $g_i(\boldsymbol{y})$, is a function of the observation vector that leads to the decision rule of selecting $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_k$ if

$$g_k(\boldsymbol{y}) > g_l(\boldsymbol{y}) \quad \forall l \neq k. \tag{6.7}$$

For the MAP detector

$$\overline{g}_i(\boldsymbol{y}) = \Pr[\boldsymbol{\alpha}_i] p(\boldsymbol{y}|\boldsymbol{\alpha}_i); \tag{6.8}$$

this choice for discriminant function is not unique, Equation (6.7) remains valid if $g_i(\boldsymbol{y})$ is replaced with $f(g_i(\boldsymbol{y}))$ for any monotonically increasing function $f(\cdot)$. Because of the form of $p(\boldsymbol{y}|\boldsymbol{\alpha}_k)$ given in Equation (6.4), a logical choice for $f(\cdot)$ is the natural logarithm, resulting in a new discriminant function

$$g_i(\boldsymbol{y}) = \ln(\Pr[\boldsymbol{\alpha}_i]) - \left(\frac{(\boldsymbol{y}-\boldsymbol{\alpha}_i)^T(\boldsymbol{y}-\boldsymbol{\alpha}_i)}{2\sigma^2} \right). \tag{6.9}$$

Observe from Equation (6.9) that the discriminant functions consist of two terms: a squared Euclidean distance measure weighted by the noise power and the *a priori* probability. If $\Pr[\boldsymbol{\alpha}_k] = \Pr[\boldsymbol{\alpha}_l] \quad \forall k, l$ then this results in the minimum distance decision rule of the ML detector. Note that unlike the ML decision rule, knowledge of the noise power σ^2 is required for the MAP detector. It is clear from Equation (6.9) that for large values of σ^2 the decision rule is weighted heavily by the prior probabilities.

Equation (6.9) can be expanded to write the discriminant function as

$$g_i(\boldsymbol{y}) = \ln(\Pr[\boldsymbol{\alpha}_i)] - \left(\frac{\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_i - 2\boldsymbol{y}^T\boldsymbol{\alpha}_i}{2\sigma^2}\right). \tag{6.10}$$

The term $\boldsymbol{y}^T\boldsymbol{y}$ can be removed because it is common to each discriminant function, leaving a new simplified discriminant:

$$g_i(\boldsymbol{y}) = \sigma^2 \ln(\Pr[\boldsymbol{\alpha_i}]) - \frac{\boldsymbol{\alpha_i}^T\boldsymbol{\alpha_i}}{2} + \boldsymbol{y}^T\boldsymbol{\alpha_i} \tag{6.11}$$

If each of the signals have equal energy, the $\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_i$ terms can be removed from Equation (6.11), and the discriminant function can be further simplified to

$$g_i(\boldsymbol{y}) = \sigma^2 \ln(\Pr[\boldsymbol{\alpha}_i]) + \boldsymbol{y}^T\boldsymbol{\alpha_i}. \tag{6.12}$$

The decision region, $\mathcal{R}_k$, consists of the values of $\boldsymbol{y}$ for which $\boldsymbol{\alpha_k}$ maximizes (6.5). The boundaries of these regions are the $(N-1)$-dimensional hyperplanes where $\boldsymbol{y}$ satisfies

$$g_k(\boldsymbol{y}) = g_l(\boldsymbol{y}). \tag{6.13}$$

To show that the boundaries are hyperplanes, consider the discriminant functions for the general MAP case. The boundary is then given by the values of $\boldsymbol{y}$ satisfying the following equation

$$\ln(\Pr[\boldsymbol{\alpha}_k)] - \left(\frac{(\boldsymbol{y}-\boldsymbol{\alpha_k})^T(\boldsymbol{y}-\boldsymbol{\alpha_k})}{2\sigma^2}\right) = \ln(\Pr[\boldsymbol{\alpha}_l)] - \left(\frac{(\boldsymbol{y}-\boldsymbol{\alpha_l})^T(\boldsymbol{y}-\boldsymbol{\alpha_l})}{2\sigma^2}\right). \tag{6.14}$$

Expand the above and cancel common terms to obtain

$$\ln(\Pr[\boldsymbol{\alpha_k}]) - \frac{\boldsymbol{\alpha}_k^T\boldsymbol{\alpha_k} - 2\boldsymbol{y}^T\boldsymbol{\alpha}_k}{2\sigma^2} = \ln(\Pr[\boldsymbol{\alpha}_l]) - \frac{\boldsymbol{\alpha}_l^T\boldsymbol{\alpha_l} - 2\boldsymbol{y}^T\boldsymbol{\alpha}_l}{2\sigma^2}, \tag{6.15}$$

and finally collect the $\boldsymbol{y}$ terms to obtain

$$\boldsymbol{y}^T(\boldsymbol{\alpha_k} - \boldsymbol{\alpha_l}) = \frac{1}{2}\boldsymbol{\alpha_k}^T\boldsymbol{\alpha_k} - \frac{1}{2}\boldsymbol{\alpha_l}^T\boldsymbol{\alpha_l} - \sigma^2 \ln\left(\frac{\Pr[\boldsymbol{\alpha_k}]}{\Pr[\boldsymbol{\alpha_l}]}\right). \tag{6.16}$$

The right hand side of Equation (6.16) is a constant scalar. The decision boundary that results is a hyperplane parallel to the vector $\boldsymbol{y}$ which satisfies $\boldsymbol{y}^T(\boldsymbol{\alpha_k} - \boldsymbol{\alpha_l}) = 0$.

If the field values are all equal energy, the discriminant functions in (6.12) can be used, resulting in boundaries defined by

$$\boldsymbol{y}^T(\boldsymbol{\alpha_k} - \boldsymbol{\alpha_l}) = \sigma^2 \ln\left(\frac{\Pr[\boldsymbol{\alpha_l}]}{\Pr[\boldsymbol{\alpha_k}]}\right). \tag{6.17}$$

There are two special cases which guarantee reachability of all decisions for some specific value of $\boldsymbol{y}$ (a value of $\boldsymbol{y}$ exists such that decision $\hat{\boldsymbol{\alpha}}_l$ is made $\forall l$). The first case is equal energy symbols; the second is equal probability sequences. In the general case of unequal energy symbols and unequal probabilities, there may be some decisions that will never be made no matter what the value of the observation vector $\boldsymbol{y}$.

For the equal energy case:

$$\boldsymbol{y}^T(\boldsymbol{\alpha_k} - \boldsymbol{\alpha_l}) \underset{H_l}{\overset{H_k}{\gtrless}} C \tag{6.18}$$

Let $\boldsymbol{y} = B\boldsymbol{\alpha}_k$, a scaler B can always be found such that decision k is made.

For the equal probability case, the decision rule results in the minimum distance decision. Setting the observation vector to $\boldsymbol{\alpha}_k$ will result in a distance of zero and decision k will be made.

Figure 6.1 illustrates the decision regions for the two symbol case in $\mathbf{R}^2$. Notice the direction the decision boundary moves as the energy in a symbol is increased or the prior probability is changed.

Some observations:

- If $\Pr[\boldsymbol{\alpha_k}] = \Pr[\boldsymbol{\alpha_l}] \quad \forall k, l$, the decision rule is the minimum distance decision rule of the ML detector, and an estimate of the noise power is no longer required.

- For large σ^2, the decision rule is heavily weighted by the prior probabilities.

- The M decision regions are separated by $(N-1)$-dimensional hyperplanes.

- For the general case, with no restriction for equal energy signals or equal prior probabilities, there may be decisions which are not made no matter what the value of $\boldsymbol{y}$. If either of the following conditions are true, $\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l \quad \forall l, k$ or $\Pr(\boldsymbol{\alpha}_k) =$

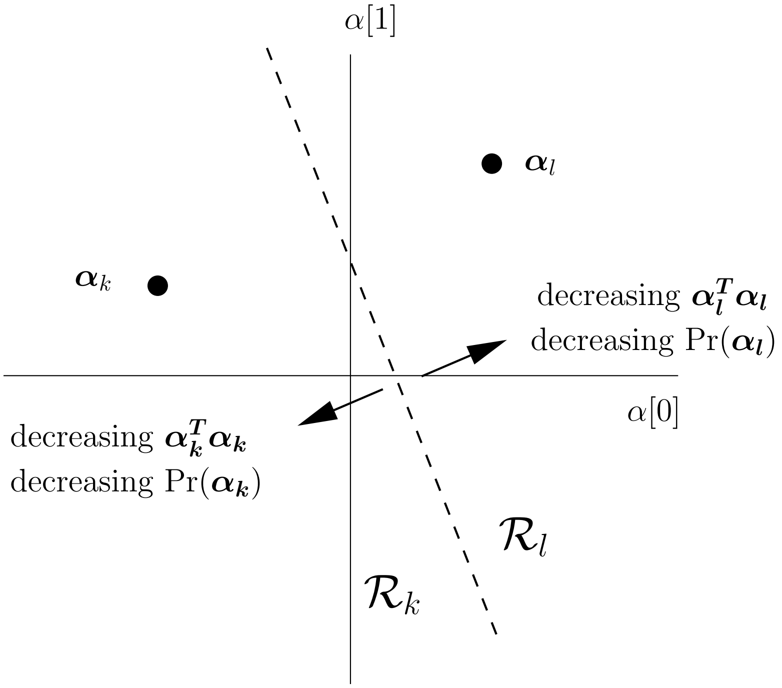

$\alpha[1]$
$\boldsymbol{\alpha}_l$
$\boldsymbol{\alpha}_k$
decreasing $\boldsymbol{\alpha_l^T \alpha_l}$
decreasing $\Pr(\boldsymbol{\alpha_l})$
$\alpha[0]$
decreasing $\boldsymbol{\alpha_k^T \alpha_k}$
decreasing $\Pr(\boldsymbol{\alpha_k})$
$\mathcal{R}_l$
$\mathcal{R}_k$

Figure 6.1: Decision regions for $\mathbf{R}^2$

$\Pr(\boldsymbol{\alpha}_l) \neq 0 \quad \forall l, k$, then there will always be a value of $\boldsymbol{y}$ that will lead to decision $\boldsymbol{\alpha_i}$.

3. Probability of Error

In this section the probability of field error for the MAP detector is derived, then a useful bound is developed that is more practical to calculate. An error occurs when the decision $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_k$ is made when $\boldsymbol{\alpha}_l, \quad l \neq k$ was sent. The probability of error is described by the following equation:

$$\begin{aligned}\Pr(\text{error}) &= \sum_k \sum_{l \neq k} \Pr(\boldsymbol{y} \in \mathcal{R}_k | \boldsymbol{\alpha}_l) \Pr[\boldsymbol{\alpha_l}] \\ &= \sum_k \sum_{l \neq k} \int_{\mathcal{R}_k} p(\boldsymbol{y}|\boldsymbol{\alpha_l}) \Pr[\boldsymbol{\alpha_l}] \, \mathrm{d}\boldsymbol{y}\end{aligned} \tag{6.19}$$

where $\mathcal{R}_k$ is the region where decision k is made. The probability of error can also be written as $1 - \Pr(\text{correct})$:

$$\begin{aligned}\Pr(\text{error}) &= 1 - \Pr(\text{correct}) \\ &= 1 - \sum_k \Pr(\boldsymbol{y} \in \mathcal{R}_k | \boldsymbol{\alpha_k}) \Pr(\boldsymbol{\alpha_k}) \\ &= 1 - \sum_k \int_{\mathcal{R}_k} \Pr(\boldsymbol{y}|\boldsymbol{\alpha_k}) \Pr(\boldsymbol{\alpha_k})\, \mathrm{d}\boldsymbol{y}\end{aligned} \tag{6.20}$$

Both (6.19) and (6.20) involve multiple $(N-1) - fold$ integrals over decision regions, resulting in complicated integration limits. A simplification can be found by considering the pairwise union upper bound.

To develop the upper bound, begin by rewriting the probability of error as

$$\Pr(\text{error}) = \sum_k \sum_{l \neq k} \Pr[g_l(\boldsymbol{y}) > g_k(\boldsymbol{y}) | \boldsymbol{\alpha_k}] \Pr(\boldsymbol{\alpha_k}) - \text{all the joint terms} \tag{6.21}$$

where each $\Pr[g_l(\boldsymbol{y}) > g_k(\boldsymbol{y})|\boldsymbol{\alpha}_k]$ term is the pairwise probability of deciding $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_l$ over $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_k$ given $\boldsymbol{\alpha}_k$ was sent. The joint terms are the intersecting areas that get accounted for multiple times from only considering pairwise decisions.

Calculation of $\Pr[g_l(\boldsymbol{y}) > g_k(\boldsymbol{y})|\boldsymbol{\alpha}_k]$ still involves an $(N-1) - fold$ integral, but the complexity of the decision region boundary is reduced to that of a single hyperplane defined by (6.16). The simplified decision region, along with the Gaussian form of $p(\boldsymbol{y}|\boldsymbol{\alpha})$, allows the transformation of the problem to a one-dimensional integral.

We can now use the equation for the received vector $\boldsymbol{y}$ (6.3) in the decision boundary defined in (6.16) to obtain

$$\begin{aligned}(\boldsymbol{\alpha}_k + \boldsymbol{z})^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l) &= \frac{1}{2}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k - \frac{1}{2}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l - \sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right] \\ 2\boldsymbol{z}^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l) &= -\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l + 2\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_l - 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right] \\ &= -(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l) - 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]\end{aligned} \tag{6.22}$$

The noise vector $\boldsymbol{z}$ from Equation (6.3) is a vector of zero-mean i.i.d. Gaussian random variables with covariance $\sigma^2\boldsymbol{I}$. The random variable $2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)^T\boldsymbol{z}$, is therefore zero-mean with variance $\sigma_1^2 = 4\sigma^2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)$.[1] The pairwise probability of making decision $\hat{\boldsymbol{\alpha}}_l$ given $\boldsymbol{\alpha}_k$ was sent is

$$\begin{aligned}
\Pr[g_l(\boldsymbol{y}) > g_k(\boldsymbol{y})|\boldsymbol{\alpha}_k] &= \int_{\mathcal{R}'_k} p(\boldsymbol{y}|\boldsymbol{\alpha}_k)\,\mathrm{d}\boldsymbol{y} \\
&= \Pr\left[2\boldsymbol{z}^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l) < -\left(d_{lk}^2 + 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]\right)\right] \\
&= \Pr\left[\frac{2\boldsymbol{z}^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)}{2\sigma d_{lk}} < -\frac{d_{lk} + 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]}{2\sigma d_{lk}}\right] \\
&= \Pr\left[\frac{2\boldsymbol{z}^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)}{2\sigma d_{lk}} > +\frac{d_{lk}^2 + 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]}{2\sigma d_{lk}}\right] \\
&= Q\left(\frac{d_{lk}^2 + 2\sigma^2 \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]}{2\sigma d_{lk}}\right) \\
&= Q\left(\frac{d_{lk}}{2\sigma} + \frac{\sigma}{d_{lk}} \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]\right)
\end{aligned} \tag{6.23}$$

Where $\mathcal{R}'_k$ is the region for decision $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_k$ when only two decisions are considered; the other decision being $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_l$. The term d_{lk} is given by $d_{lk} = \sqrt{(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)}$.

This result is used in (6.21) to obtain the union upper bound[2] for the probability of field error $P_{fe(MAP)}$:

$$P_{fe(MAP)} \leq \sum_k \Pr(\boldsymbol{\alpha}_k) \sum_{l \neq k} Q\left(\frac{d_{lk}}{2\sigma} + \frac{\sigma}{d_{lk}} \ln\left[\frac{\Pr(\boldsymbol{\alpha}_k)}{\Pr(\boldsymbol{\alpha}_l)}\right]\right) \tag{6.24}$$

For high SNR the "joint terms", neglected in developing this bound, become small and this bound becomes tight.

A less complex upper bound may be found by considering the minimum distance, d_{min}, as a worst case. Form the M-ary upper bound [8] by considering all of the sequences equally likely and recognizing that all of the distance terms d_{lk} in Equation (6.24) will be

[1] Given a random vector $\boldsymbol{x}$ with covariance matrix $\boldsymbol{R_x}$ the vector $\boldsymbol{y}$ given by the linear transformation $\boldsymbol{Ax}$ has autocorrelation function $\boldsymbol{R_y} = \boldsymbol{A R_x A}^{*T}$ [103]. $4\sigma^2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)^T(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_l)$

[2] The union upper bound says $\Pr\left(\bigcup_i A_i\right) \leq \sum_i \Pr(A_i)$ [11]

greater than or equal to d_{min}. The M-ary upper bound is then given by

$$(M-1)Q\left(\sqrt{\frac{{d_{min}}^2}{2N_0}}\right), \tag{6.25}$$

where d_{min} is the minimum Euclidean distance between any of the field values. Notice that for this bound there are still only M possible field values transmitted rather than 2^N, but the field values are assumed equally likely. Although this bound is simple to calculate, it is not very tight for large values of M. Hereafter, this bound (6.25) will not be used.

4. Matrix Formulation

For the purposes of simulation of field error rate and calculation of the upper bound, it is convenient to write the above equations in matrix notation. By writing the equations in a matrix format, some of the loops can be eliminated allowing the use of optimized matrix libraries.

Consider the transmission of L fields. The received signal Equation (6.3) can be extended as a matrix of received fields as

$$\boldsymbol{Y} = \boldsymbol{B} + \boldsymbol{Z}, \tag{6.26}$$

where $\boldsymbol{B}$ is an $N \times L$ matrix of the L particular transmitted column vectors $\boldsymbol{\alpha}$, and N is the number of symbols in each vector $\boldsymbol{\alpha}$. Each column vector $\boldsymbol{\alpha}$ is drawn from the $M < 2^N$ values for the vector $\boldsymbol{\alpha}$ with non-negligible *a priori* probabilities ($p_k > \epsilon$). The matrix $\boldsymbol{Z}$ is an $N \times L$ matrix of the L noise column vectors $\boldsymbol{z}$.

The discriminant functions given in 6.11 can be calculated using matrix notation as follows:

$$\boldsymbol{G} = \sigma^2 \ln(\boldsymbol{p}\boldsymbol{o}_L^T) - \frac{1}{2}(\boldsymbol{o}_N^T \boldsymbol{A}_{\alpha\alpha^*})^T \boldsymbol{o}_L^T + \boldsymbol{A}^T\boldsymbol{Y}), \tag{6.27}$$

where $\ln(\cdot)$ of a matrix is an element-by-element natural log. The vector $\boldsymbol{o}_j$ is a length j column vector of ones. The vector $\boldsymbol{p}$ is an $M \times 1$ vector of the *a priori* probabilities for each possible field $\Pr(\boldsymbol{\alpha}_k) = p_k$. The matrix $\boldsymbol{A}_{\alpha\alpha^*}$ is an $N \times M$ matrix of column vectors, where the i^{th} column of $\boldsymbol{A}_{\alpha\alpha^*}$ is given by $\begin{bmatrix}\alpha_i[0]\alpha_i^*[0] & \alpha_i[1]\alpha_i^*[1] & \cdots & \alpha_i[N-1]\alpha_i^*[N-1]\end{bmatrix}^T$. The matrix $\boldsymbol{A}$ is an $N \times M$ matrix of the possible transmitted signals $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{\alpha}_0 & \boldsymbol{\alpha}_1 & \ldots & \boldsymbol{\alpha}_{M-1} & \end{bmatrix}$.

Table 6.1: Example distributions for $\boldsymbol{\alpha}$

field	field values separation of 1	field values separation of 10	p_k	$\ln(p_k)$
$\boldsymbol{\alpha_0}$	n - 5	n-50	0.011	-4.4
$\boldsymbol{\alpha_1}$	n - 4	n-40	0.003	-5.7
$\boldsymbol{\alpha_2}$	n - 3	n-30	0.005	-5.4
$\boldsymbol{\alpha_3}$	n - 2	n-20	0.015	-4.2
$\boldsymbol{\alpha_4}$	n - 1	n-10	0.109	-2.2
$\boldsymbol{\alpha_5}$	n	n	0.519	-0.7
$\boldsymbol{\alpha_6}$	n + 1	n+10	0.291	-1.2
$\boldsymbol{\alpha_7}$	n + 2	n+20	0.031	-3.5
$\boldsymbol{\alpha_8}$	n + 3	n+30	0.004	-5.4
$\boldsymbol{\alpha_9}$	n + 4	n+40	0.003	-5.9
$\boldsymbol{\alpha_{10}}$	n + 5	n+50	0.008	-4.8

The resulting $M \times L$ matrix $\boldsymbol{G}$ can be used to determine the field-based MAP estimate for $\boldsymbol{\alpha}$. The maximum value in each column of $\boldsymbol{G}$ is associated with the estimate for $\boldsymbol{\alpha}$ from the MAP decision rule.

5. Results

Consider the case of 11 likely field values shown in Table 6.1. Each field is 28 bits in length, ($N = 28$). With 28 symbols per field, there are 2^{28} possible field values; for this example we consider $M = 11$ of them to be likely. In practice this distinction is made by considering the possible field values to be those whose *a priori* probability $p_k > \epsilon$, for some selected threshold ϵ. By making this selection there is a residual error rate of

$$P_{res} = \sum_{k, \forall \rho_k < \epsilon} \rho_k. \tag{6.28}$$

The amount of tolerable residual error will ultimately determine the maximum practical value for ϵ.

Each field has *a priori* probability and values as shown in the Table 6.1. This section presents results for two cases. In the first case, the likely transmitted values are represented as consecutive numbers. In the second, the values are separated by steps of ten. In each case, the transmitted symbols are mapped to the field values using binary encoding. In each case, the Euclidean distance between field values depends on a nominal initial value n.

Figures 6.2b and 6.3b show the field error performance of the field-based MAP detector compared to the field error performance of a ML symbol-by-symbol detector for a field value separation of one and ten respectively. The performance of the two detectors is found by simulation. The upper bound is from Equation (6.24). This bound and the coherently detected antipodal signaling curves are analytic results.

Figure 6.2b shows the field error performance for the field-based MAP detector versus the field error performance for a ML symbol-by-symbol detector. In this example the values are separated by a value of one, with distribution illustrated in Figure 6.2a. Notice in this case the field error rate of the field-based MAP detector is close to that of the bit error rate for the optimum detector for antipodal signaling. The field-based MAP detector performs significantly better compared to the field error rate performance of the symbol-by-symbol detector; providing approximately 2 – 6 dB of gain.

Figure 6.3b shows the field error performance for the field-based MAP detector versus the field error performance for a ML symbol-by-symbol detector. In this example the values are separated by a value of ten with distribution illustrated in Figure 6.3a. Because of the increased separation between field values, the field error rate performance is superior to that of the bit error rate performance of the optimal detector for antipodal signals. This results in better field error rate performance than the corresponding symbol-by-symbol detector; providing approximately 6 – 8 dB of gain.

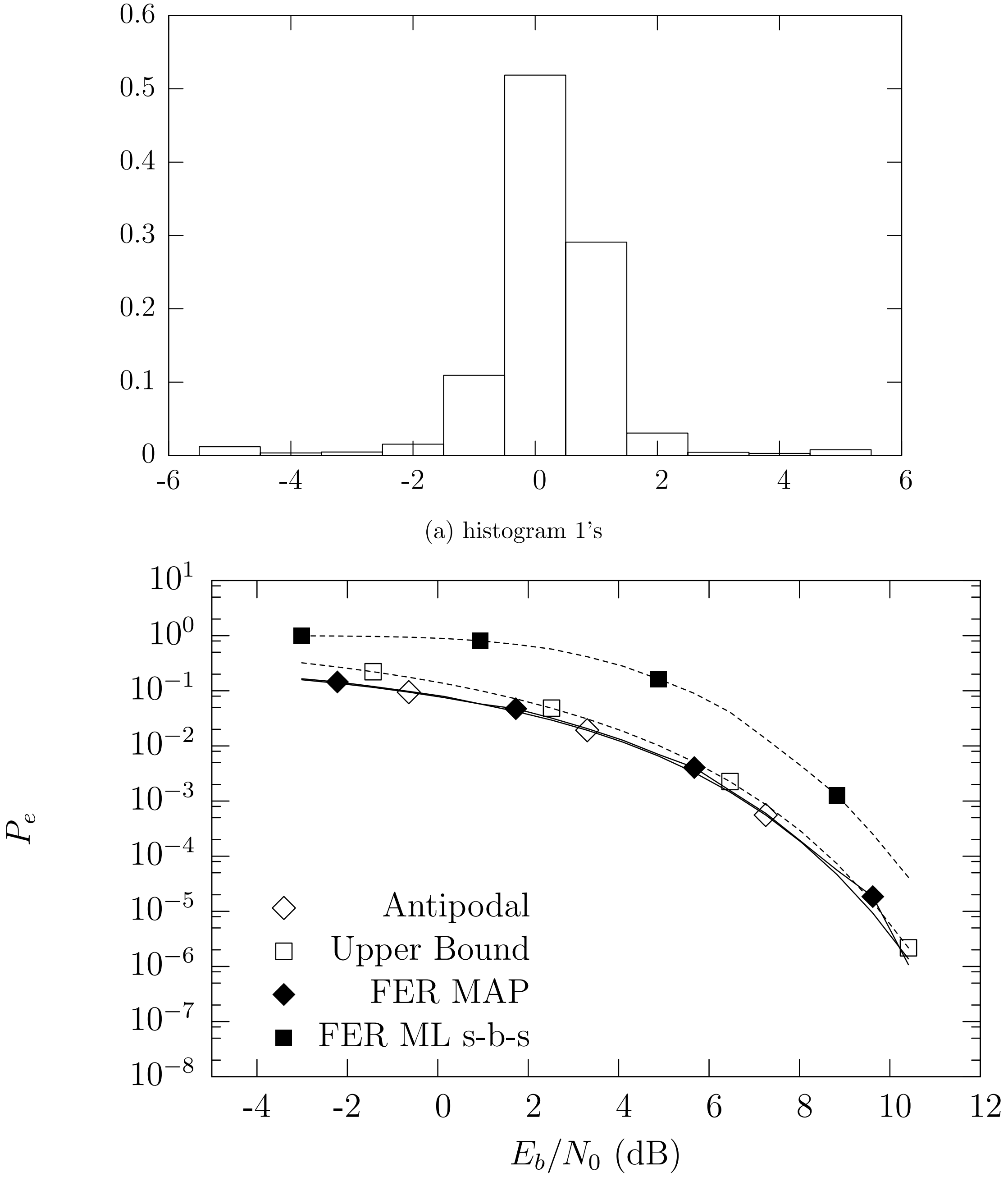
The field error performance of the ML symbol-by-symbol detector is given by

$$P_{fe(s-b-s)} = 1 - (1 - P_{se})^N, \tag{6.29}$$

where P_{se} is the probability of individual symbol error. The field error upper bound is the union upper bound (6.24).

The field values used in the bounds and the performance curves are generated by randomly selecting a number $n \in \{M/2, 2^N - M/2 - 1\}$, and then generating the appropriate $M = 11$ field values according to Table 6.1. Assuming a symbol can take on one of two values, a field of length N can represent 2^N different values. These 2^N values are mapped to the integers from 0 to $2^N - 1$. It should be apparent that the distances d_{lk} depends on the values a field can represent.

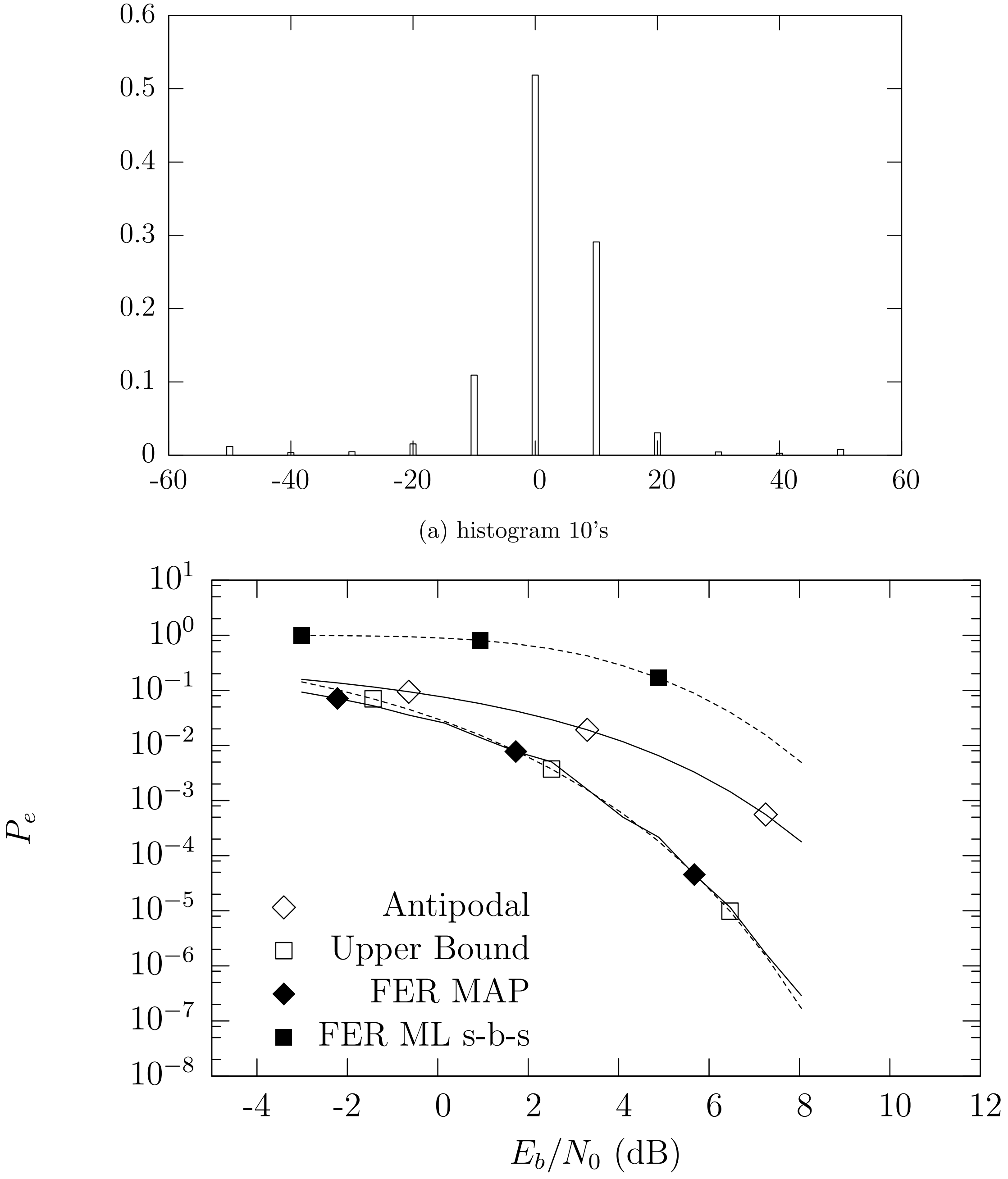
Notice the significant difference in performance for the two cases. The reason for the difference in performance between the two is the pairwise distances for the data used in Figure 6.3b are greater than the pairwise distances for the data used in Figure 6.2b.



(a) histogram 1's

(b) Error Rate 1's

Figure 6.2: Field-Based MAP for 28 Bit Field Values Separated by 1



(a) histogram 10's

(b) Error Rate 10's

Figure 6.3: Field-Base MAP for 28 Bit Field Values Separated by 10

Consider the case $N = 10$, $M = 4$ and $n = 20$ and a field separation of ten. This will lead to the pairwise distances shown in Table 6.2. Now consider the same case with a field separation of 1. This leads to the pairwise distances shown in Table 6.3. Notice the difference in the minimum distance, d_{min}, for the two cases. The field error performance of the receiver will increase as the Euclidean distance between the pairs increases. Assuming equally likely field values, the smallest pairwise distance d_{min} will have the greatest contribution to the probability of error.

Table 6.2: Distances separation of 10

pair	Euclidean Distance
$d_{10,20}$	1.2649
$d_{10,30}$	0.8944
$d_{10,40}$	0.8944
$d_{20,30}$	0.8944
$d_{20,40}$	1.2649
$d_{30,40}$	1.2649

Table 6.3: Distances separation of 1

pair	Euclidean Distance
$d_{19,20}$	1.0954
$d_{19,21}$	0.8944
$d_{19,22}$	0.8944
$d_{20,21}$	0.6325
$d_{20,22}$	0.6325
$d_{21,22}$	0.8944

B. JOINT MAP DETECTOR (SYNCHRONOUS SIGNALS)

In the next chapter, Chapter VII, the Field-based MAP detector is extended to handle multiple asynchronous cochannel signals. Below the Field-based MAP detector is extended to the special case of *synchronous* cochannel reception. Synchronous reception is when the symbols from each source have identical start and stop times from the perspective of the receiver.

Consider two signals each with fields of length N, synchronized in time. As before, the k^{th} field from the i^{th} signal can take on one of $M_{i,k}$ values. The simultaneous synchronous reception of signal 0 and signal 1 with $M_{0,k}$ and $M_{1,k}$ possible values respectively will, in general, lead to $M_{0,k} \times M_{1,k}$ combinations. Note the implicit assumption that the underlying waveforms used to generate the symbol sequences have possibly unequal signal powers, and a cross-correlation coefficient ρ, given by (6.30), with absolute value less than 1, $|\rho| < 1$. Now the decision problem is deciding which of $M_{0,k} \times M_{1,k}$ possible field pairs was sent. The performance of the resulting detector will depend on the possible values of the fields for each signal. The pairwise distances between these field-pairs may be much smaller than the distances for the single case.

1. Model

Consider the synchronous joint reception model defined in Chapter II. Recall the cross-correlation coefficient between the two signaling waveforms defined as

$$\rho = \frac{\int \psi_0(t)\psi_1^*(t)\,\mathrm{d}t}{\sqrt{\int \psi_0(t)\psi_0^*(t)\,\mathrm{d}t}\sqrt{\int \psi_1(t)\psi_1^*(t)\,\mathrm{d}t}}. \tag{6.30}$$

For notational convenience, let $\boldsymbol{y'}$ be an interleaved vector of each matched filter output. For the two channel case:

$$\boldsymbol{y'} = \Big[y_0[0] \quad y_1[0] \quad y_0[1] \quad y_1[1] \quad \cdots \quad y_0[N-1] \quad y_1[N-1]\Big]^T \tag{6.31}$$

In this section the prime is used to represent an interleaved vector. The vectors $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ are the outputs of unit-energy filters ($\int h(t)h^*(t)\,\mathrm{d}t = 1$) matched to the first and second signal respectively, and sampled at the symbol period. The output vectors are

$$\boldsymbol{y}_0 = \boldsymbol{\alpha}_0 + \rho\boldsymbol{\alpha}_1 + \boldsymbol{z}_0, \tag{6.32}$$

and

$$\boldsymbol{y}_1 = \rho\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 + \boldsymbol{z}_1, \tag{6.33}$$

where the vectors $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_1$ are the transmitted values for signal 0 and signal 1 respectively, and the elements of $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$ are zero-mean i.i.d. Gaussian random variables each with variance σ^2. Note that the variance of the random variables contained in the vectors $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$ are equal because they are both obtained from filtering the same AWGN process with unit-energy filters. The cross-correlation of $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$ is given by $\mathcal{E}\{\boldsymbol{z}_0\boldsymbol{z}_1^{*T}\} = \sigma^2\rho\boldsymbol{I}$, where ρ is the cross-correlation coefficient.

The covariance matrix of the noise component of the vector $\boldsymbol{y'}$ defined in Equation (6.31) is Σ given by

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \rho & 0 & 0 & \dots & 0 & 0 \\ \rho & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \rho & \dots & 0 & 0 \\ 0 & 0 & \rho & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & \rho \\ 0 & 0 & 0 & 0 & \dots & \rho & 1 \end{bmatrix} \tag{6.34}$$

The interleaved observation vector from (6.31) can be re-written as:

$$\boldsymbol{y}' = \boldsymbol{\alpha}'_k + \boldsymbol{z}' \tag{6.35}$$

where the vector $\boldsymbol{\alpha}'_k$ is an interleaved vector of one of the $M_{0,j} \times M_{1,l}$ possible transmitted pairs:

$$\boldsymbol{\alpha_k}' = \begin{bmatrix} \alpha_{0,j}[0] + \rho\alpha_{1,l}[0] \\ \rho\alpha_{0,j}[0] + \alpha_{1,l}[0] \\ \alpha_{0,j}[1] + \rho\alpha_{1,l}[1] \\ \rho\alpha_{0,j}[1] + \alpha_{1,l}[1] \\ \vdots \\ \alpha_{0,j}[N-1] + \rho\alpha_{1,l}[N-1] \\ \rho\alpha_{0,j}[N-1] + \alpha_{1,l}[N-1] \end{bmatrix} \tag{6.36}$$

The vector $\boldsymbol{\alpha'_k}$ is built from one of the pairs of vectors described in

$$\Big\{(\boldsymbol{\alpha}_{0,0}, \boldsymbol{\alpha}_{1,0}),\ \ (\boldsymbol{\alpha}_{0,0}, \boldsymbol{\alpha}_{1,1}),\ \ \cdots,\ \ (\boldsymbol{\alpha}_{0,0}, \boldsymbol{\alpha}_{1,M_1-1}),\ \ (\boldsymbol{\alpha}_{0,1}, \boldsymbol{\alpha}_{1,0}),\ \ \cdots,\ \ (\boldsymbol{\alpha}_{0,M_0-1}, \boldsymbol{\alpha}_{1,M_1-1})\Big\} \tag{6.37}$$

The conditional PDF of the length $2N$ vector $\boldsymbol{y}'$ is then multivariate Gaussian given by

$$p(\boldsymbol{y}'|\boldsymbol{\alpha}'_k) = \frac{1}{(2\pi)^{2N/2}|\Sigma|^{1/2})} \exp\left(\frac{-(\boldsymbol{y}' - \boldsymbol{\alpha}'_k)^T\Sigma^{-1}(\boldsymbol{y}' - \boldsymbol{\alpha}'_k)}{2}\right) \tag{6.38}$$

2. Discriminant Functions

The discriminant function, as introduced in Section VI.A.2, for the conditional PDF given in Equation (6.38) is given by [102]

$$\begin{aligned} \tilde{g}_i(\boldsymbol{y}') &= \ln(\Pr[\boldsymbol{\alpha}'_i]) - \left(\frac{(\boldsymbol{y}' - \boldsymbol{\alpha}'_i)^T\Sigma^{-1}(\boldsymbol{y}' - \boldsymbol{\alpha}'_i)}{2}\right) \\ &= \ln(\Pr[\boldsymbol{\alpha}'_i]) - \left(\frac{(\boldsymbol{y}')^T\Sigma^{-1}\boldsymbol{y}' - 2(\boldsymbol{\alpha}'_i)^T\Sigma^{-1}\boldsymbol{y}' + (\boldsymbol{\alpha}'_i)^T\Sigma^{-1}\boldsymbol{\alpha}'_i}{2}\right) \end{aligned} \tag{6.39}$$

The second line of (6.39) follows from the identity $\Sigma^{-1} = (\Sigma^{-1})^{*T}$ from the complex conjugate symmetry of the covariance matrix. The term $(\boldsymbol{y}' - \boldsymbol{\alpha}'_i)^T\Sigma^{-1}(\boldsymbol{y}' - \boldsymbol{\alpha}'_i)$ is the square of the Mahalanobis distance [11] for $\boldsymbol{y}'$ and $\boldsymbol{\alpha}'_i$. The terms $\boldsymbol{y}'^T\Sigma^{-1}\boldsymbol{y}'$ are common to each discriminant; they may be removed leading to new a simplified discriminant:

$$g_i(\boldsymbol{y}') = \ln(\Pr[\boldsymbol{\alpha}'_i]) - \frac{1}{2}(\boldsymbol{\alpha}'_i)^T\Sigma^{-1}\boldsymbol{\alpha}'_i + (\boldsymbol{\alpha}'_i)^T\Sigma^{-1}\boldsymbol{y}' \tag{6.40}$$

As in the single-channel case, the decision boundaries are still hyperplanes. Unlike the single-channel case, the hyperplane defining the decision boundary will not, in general, be perpendicular to the vector connecting $\boldsymbol{\alpha}'_l$ and $\boldsymbol{\alpha}'_k$.

3. Probability of Error

The development of the probability of error calculation closely follows that for the single-channel case developed in Section VI.A.3. The union bound is used in an identical fashion. The only difference is in the calculation of the pairwise probability of error, $\Pr[g_l(\boldsymbol{y}') > g_k(\boldsymbol{y}')|\boldsymbol{\alpha}'_k]$. To derive the probability of error, begin by writing out the description of the decision boundary $g_l(\boldsymbol{y}') = g_k(\boldsymbol{y}')$.

$$\ln(\Pr[\boldsymbol{\alpha}'_l]) - \frac{(\boldsymbol{\alpha}'_l)^T\Sigma^{-1}\boldsymbol{\alpha}'_l}{2} + (\boldsymbol{\alpha}'_l)^T\Sigma^{-1}\boldsymbol{y}' = \ln(\Pr[\boldsymbol{\alpha}'_k]) - \frac{(\boldsymbol{\alpha}'_k)^T\Sigma^{-1}\boldsymbol{\alpha}'_k}{2} + (\boldsymbol{\alpha}'_k)^T\Sigma^{-1}\boldsymbol{y}' \tag{6.41}$$

then collect the $\boldsymbol{y'}$ terms on the right hand side.

$$\begin{aligned}
(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{y'} - (\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{y'} &= -\ln(\Pr[\boldsymbol{\alpha}_l']) + \ln(\Pr[\boldsymbol{\alpha}_k']) + \frac{(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{\alpha}_l'}{2} - \frac{(\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{\alpha}_k'}{2} \\
(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{y'} - (\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{y'} &= \ln\left(\frac{\Pr[\boldsymbol{\alpha}_k']}{\Pr[\boldsymbol{\alpha}_l']}\right) + \frac{(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{\alpha}_l'}{2} - \frac{(\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{\alpha}_k'}{2}
\end{aligned} \tag{6.42}$$

The decision boundary is then found by then assuming $\boldsymbol{\alpha}_k'$ was sent and replacing the interleaved observation vector, $\boldsymbol{y'}$, with the right hand side of Equation (6.35).

$$\begin{aligned}
(\boldsymbol{\alpha}_l')^T\Sigma^{-1}(\boldsymbol{\alpha_k}' + \boldsymbol{z}') - (\boldsymbol{\alpha}_k')^T\Sigma^{-1}(\boldsymbol{\alpha_k}' + \boldsymbol{z}') &= \ln\left(\frac{\Pr[\boldsymbol{\alpha}_k']}{\Pr[\boldsymbol{\alpha}_l']}\right) + \frac{(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{\alpha}_l'}{2} - \frac{(\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{\alpha}_k'}{2} \\
(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{z}' &= \ln\left(\frac{\Pr[\boldsymbol{\alpha}_k']}{\Pr[\boldsymbol{\alpha}_l']}\right) + \frac{(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{\alpha}_l'}{2} + \frac{(\boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{\alpha}_k'}{2} \\
&\quad -(\boldsymbol{\alpha}_l')^T\Sigma^{-1}\boldsymbol{\alpha}_k' \\
(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{z}' &= \ln\left(\frac{\Pr[\boldsymbol{\alpha}_k']}{\Pr[\boldsymbol{\alpha}_l']}\right) + \frac{(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')}{2}
\end{aligned} \tag{6.43}$$

Since $\boldsymbol{z}'$ has covariance matrix Σ, the random variable $(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{z}'$ has variance $\sigma_1^{2'} = (\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^*$.[3]

[3] Given a random vector $\boldsymbol{x}$ with covariance matrix $\boldsymbol{R_x}$ the vector $\boldsymbol{y}$ given by the linear transformation $\boldsymbol{Ax}$ has autocorrelation function $\boldsymbol{R_y} = \boldsymbol{AR_xA}^{*T}$ [103].

$$(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1}\boldsymbol{\Sigma}((\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\Sigma^{-1})^{*T} = (\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')^T\boldsymbol{\Sigma}^{-1*}(\boldsymbol{\alpha}_l' - \boldsymbol{\alpha}_k')$$

Following the same methods used in the single-channel case in Section VI.A.3 the probability of pairwise error is given by

$$\begin{aligned}
\Pr[g_l(\boldsymbol{y'}) > g_k(\boldsymbol{y'}|\boldsymbol{\alpha}'_k)] &= \int_{\mathcal{R}'_k} p(\boldsymbol{y'}|\boldsymbol{\alpha'}_k)\,\mathrm{d}\boldsymbol{y'} \\
&= \Pr\left[(\boldsymbol{\alpha}'_l - \boldsymbol{\alpha}'_k)^T\Sigma^{-1}\boldsymbol{z'} < -\left(\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M^2(l,k)}{2}\right)\right] \\
&= \Pr\left[(\boldsymbol{\alpha}'_l - \boldsymbol{\alpha}'_k)^T\Sigma^{-1}\boldsymbol{z'} < -\frac{\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M^2(l,k)}{2}}{d_M(l,k)}\right] \\
&= \Pr\left[\frac{(\boldsymbol{\alpha}'_l - \boldsymbol{\alpha}'_k)^T\Sigma^{-1}\boldsymbol{z'}}{d_M(l,k)} > +\frac{\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M^2(l,k)}{2}}{d_M(l,k)}\right] \\
=& Q\left(\frac{\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M^2(l,k)}{2}}{d_M(l,k)}\right) \\
=& Q\left(\frac{1}{d_M(l,k)}\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M(l,k)}{2}\right)
\end{aligned} \tag{6.44}$$

Where $d_M^2(l,k)$ is the squared Mahalanobis distance [11] between $\boldsymbol{\alpha}'_l$ and $\boldsymbol{\alpha}'_k$ given by

$$d_M^2(l,k) = (\boldsymbol{\alpha}'_l - \boldsymbol{\alpha}'_k)^T\Sigma^{-1}(\boldsymbol{\alpha}'_l - \boldsymbol{\alpha}'_k) \tag{6.45}$$

Notice if $\Sigma = \sigma^2\boldsymbol{I}$ then $d_M^2(l,k) = d_{lk}^2/\sigma^2$, and Equation (6.44) reduces to Equation (6.23).

Following the method of Section VI.A.3, the result in (6.21) can be used to obtain the union upper bound for the probability of field error for joint signals $P_{fe(JMAP)}$:

$$P_{fe(JMAP)} \leq \sum_k \Pr(\boldsymbol{\alpha'}_k)\sum_{l\neq k} Q\left(\frac{1}{d_M(l,k)}\ln\left(\frac{\Pr[\boldsymbol{\alpha}'_k]}{\Pr[\boldsymbol{\alpha}'_l]}\right) + \frac{d_M(l,k)}{2}\right) \tag{6.46}$$

4. Matrix Formulation

As in the single-channel case, it is often useful to develop a matrix formulation of the equations to allow the use of vector math libraries for computation. The matrix formulation is developed for Equation (6.40), in a similar fashion to that of the single-channel case (6.27).

The received signal (6.31) can be extended as a matrix of received fields as

$$\boldsymbol{Y'} = \boldsymbol{B'} + \boldsymbol{Z'} \tag{6.47}$$

Where $\boldsymbol{B'}$ is an $2N \times L$ matrix of the L transmitted column vectors $\boldsymbol{\alpha'}$, and $2N$ is the number of symbols in each vector $\boldsymbol{\alpha'}$. Each column vector $\boldsymbol{\alpha'}$ is drawn from the $M_0 M_1 < 2^{2N}$ possible values for the vector $\boldsymbol{\alpha'}$. The matrix $\boldsymbol{Z'}$ is an $2N \times L$ matrix of the L noise column vectors $\boldsymbol{z'}$.

The discriminant functions given in (6.40) can be calculated using matrix notation as follows

$$\boldsymbol{G} = \ln(\boldsymbol{p}\boldsymbol{o}_L^T) - \frac{1}{2}(\boldsymbol{o}_{2N}^T \boldsymbol{A}_{\alpha\Sigma^{-1}\alpha^*})^T \boldsymbol{o}_L^T + (\boldsymbol{A'})^T \Sigma^{-1} \boldsymbol{Y'}) \tag{6.48}$$

The vector $\boldsymbol{o}_j$ is a length j column vector of ones. The vector $\boldsymbol{p}$ is an $(M_0 M_1) \times 1$ vector of the prior probabilities $\Pr(\boldsymbol{\alpha'}_i)$. The matrix $\boldsymbol{A}_{\alpha\Sigma^{-1}\alpha^*}$ is a $2N \times (M_0 M_1)$ matrix of column vectors, where the i^{th} column of $\boldsymbol{A}_{\alpha\Sigma^{-1}\alpha^*}$ is given by $\mathrm{diag}\{\boldsymbol{\alpha'_i}\Sigma^{-1}\boldsymbol{\alpha'}^{*T}_{\boldsymbol{i}}\}$. The matrix $\boldsymbol{A'}$ is an $N \times M$ matrix of the possible transmitted signals $\boldsymbol{A'} = \begin{bmatrix} \boldsymbol{\alpha}'_0 & \boldsymbol{\alpha}'_1 & \dots & \boldsymbol{\alpha}'_{(M_0-1)(M_1-1)} & \end{bmatrix}$

The resulting $(M_0 M_1) \times L$ matrix $\boldsymbol{G'}$ can be used to determine the field-based MAP estimate for $\boldsymbol{\alpha'}$.

5. Results

Results are presented below for the Joint field-based MAP detector with two synchronous signals. The values transmitted for each signal are determined using Table 6.1 in the same manner as for the single-channel case.

Figures 6.4 and 6.5 show the field error performance of the joint field-based MAP detector compared to the field error performance of a Joint ML symbol-by-symbol detector for a field value separation of one and ten respectively. The performance of the two detectors was found by simulation; the upper bound and the coherently detected antipodal signal curves are analytic results.

Figure 6.4 shows the field error performance for the joint field-based MAP detector compared to the field error performance for a Joint ML symbol-by-symbol detector. Note that a single-channel receiver for cochannel reception is not considered; the performance of a single channel receiver is unacceptably poor for highly correlated signals of similar received power levels. In this example the values are separated by a value of one with distribution illustrated in Figure 6.2a. Field error performance curves are generated for three different values of the cross-correlation coefficient, ρ. Notice the case of $\rho = 0$

gives identical performance to that of the single-channel case presented in Figure 6.2b. If $\rho = 0$, the underlying waveforms used to transmit the information are uncorrelated and the optimal joint receiver reduces to two single-channel receivers.

Figure 6.5 shows the field error performance for the field-based MAP detector versus the field error performance for a Joint ML symbol-by-symbol detector. In this example the values are separated by a value of ten with distribution illustrated in Figure 6.3a.

The field error performance of the Joint ML symbol-by-symbol detector is given by

$$P_{fe} = 1 - (1 - P_{se})^N \tag{6.49}$$

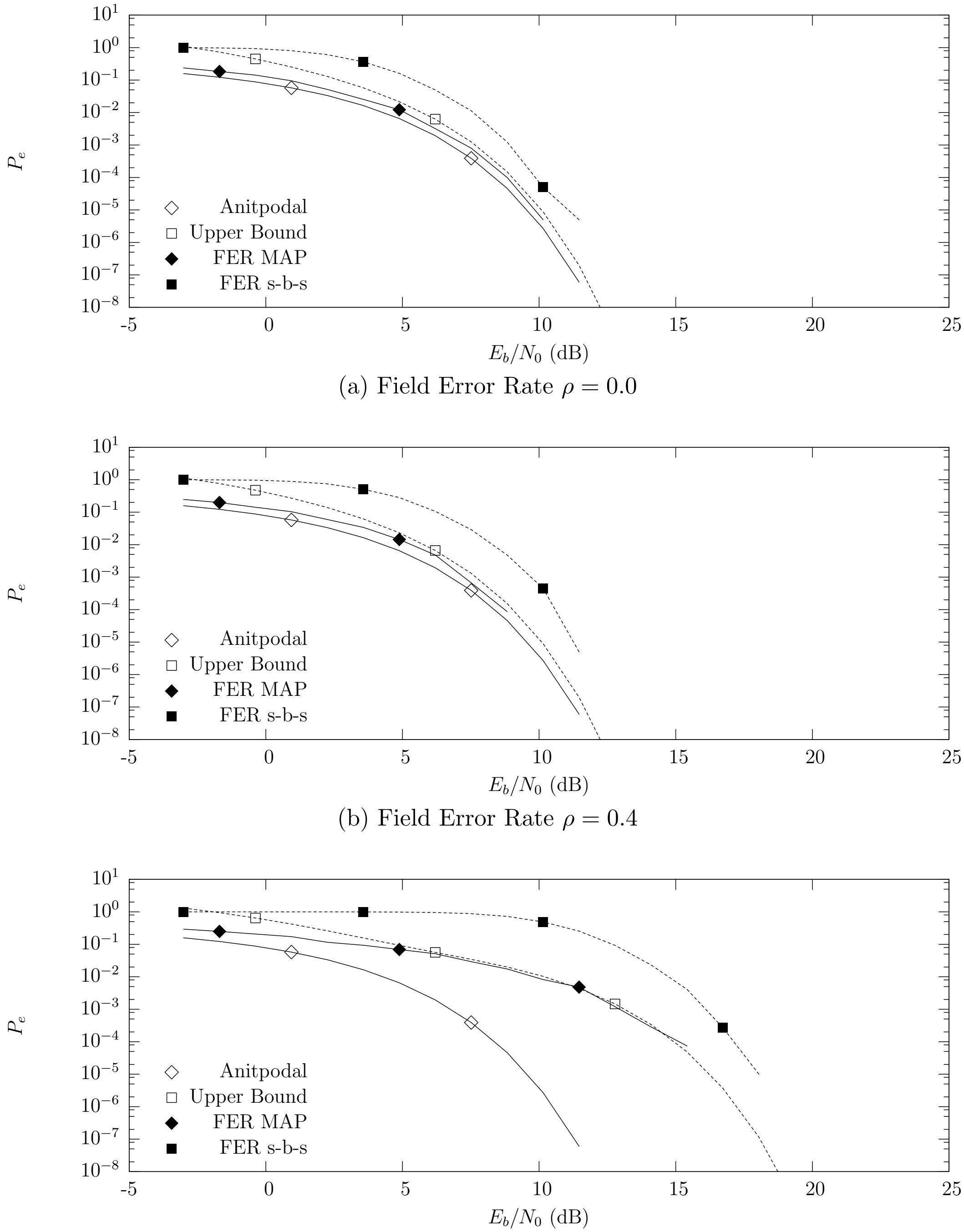
Where P_{se} is the probability of individual symbol error from a ML detector.

The field error upper bound is the union upper bound (6.46) developed in Section VI.B.3. The probability of bit error for antipodal is also included for reference. As in the case for the single-channel example, the performance will depend on the distribution of possible transmitted values for each channel.

Notice the significant difference in performance for the two cases. The reason for the difference in performance between the two is the pairwise distances for the data used in Figure 6.5 are greater than the pairwise distances for the data used in Figure 6.4.

C. COMPUTATIONAL COMPLEXITY

Here, we briefly examine the computational complexity of a brute force implementation of the MAP field detector with field length N. Calculation of the discriminants in Equation (6.11) involves $2N$ multiply and accumulate (MAC) operations. There are 2^N discriminant functions resulting in a complexity of $\approx 2^N 2N$ MACs per field. For a field length of 10 bits this amounts to approximately 20,000 MACs, for a field length of 15 this is on the order of 10^6 operations. There is an exponential dependence on field length that makes this problem unfeasible for fields consisting of many more than ten symbols.

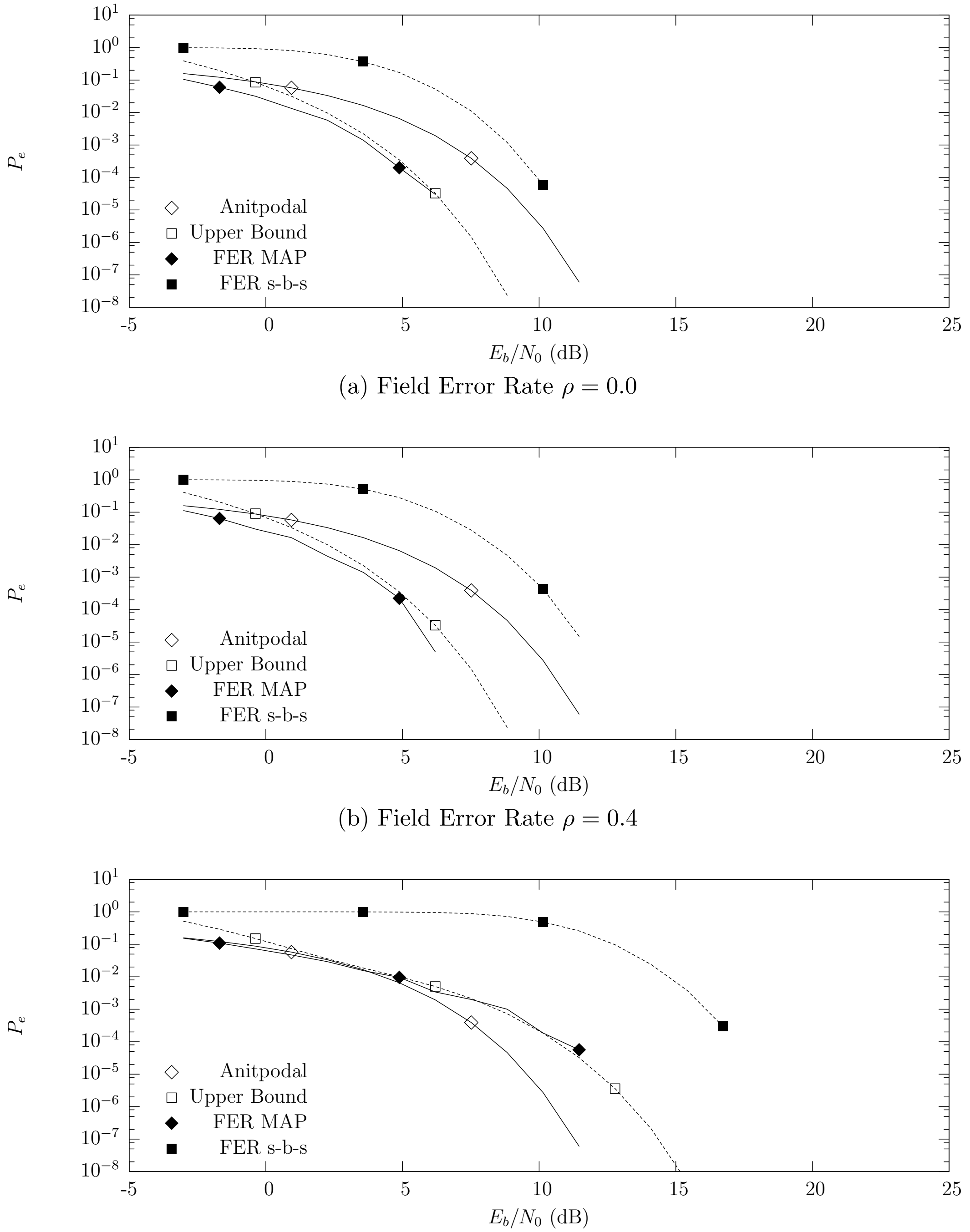


(a) Field Error Rate $\rho = 0.0$

(b) Field Error Rate $\rho = 0.4$

(c) Field Error Rate $\rho = 0.9$

Figure 6.4: Joint Field-Based MAP for 28 bit Field Values Separated by 1



(a) Field Error Rate $\rho = 0.0$

(b) Field Error Rate $\rho = 0.4$

(c) Field Error Rate $\rho = 0.9$

Figure 6.5: Joint Field-Based MAP for 28 Bit Field Values Separated by 10

One obvious reduction can be made by setting field probabilities that are negligible to zero. If only M non-negligible field probabilities remain, the complexity is reduced to $2MN$. Even with M in the thousands this remains a tractable problem.

D. CHAPTER SUMMARY

In this chapter a field-based MAP detector is developed for a vector of symbols where the *a priori* probabilities are for a field of symbols rather than the individual symbols. A gain of 2 – 8 dB in field (or sequence) error rate performance is found when *a priori* knowledge is used compared to a symbol-by-symbol ML detector. The symbols are assumed independent, i.e., no ISI and no inter-symbol dependence introduced at the transmitter.

The field-based MAP detector is then extended to multiple signals, where the fields are restricted to be synchronous. The synchronous joint field-based MAP detector is demonstrated to have performance gains of 3 – 18 dB when compared to the joint ML detector. These gains are very dependent on the *a priori* distributions.

The next chapter develops the Joint Field-based BCJR algorithm for joint field-based MAP detection using field-based *a priori* information. This algorithm allows the computation of a field-by-field MAP estimate where asynchronous cochannel signals introduce a dependency between consecutive fields. It will be seen that in order to make a decision on a particular field, all of the received data must be examined, not just the data for the current field.

VII. JOINT FIELD-BASED MAP DETECTION

In the first half of Chapter VI, the Field-Based MAP Detector is developed for an individual field of symbols. In the second half of that chapter, the Field-Based MAP Detector is extended to the Joint Field-Based MAP Detector for the special case of synchronous cochannel signals. For the special case of synchronous cochannel signals, the optimum MAP detector is developed by observing only the current field. This chapter extends the work from the second half of Chapter VI to the general case of asynchronous cochannel signals. For asynchronous signals, the optimal MAP decision can not be made by observing only the current field; the entire received vector must be used in order to make optimal MAP decisions.

The development of the Joint Field-Based MAP detector begins with a model of the multiple signal sources in a form that is compatible with the BCJR algorithm. The BCJR (or MAP algorithm) is introduced in Chapter II and detailed in Appendix B. The BCJR algorithm provides a computationally efficient method to calculate the *a posteriori* probabilities of the state transitions of a Markov chain based on an entire received vector of decision statistics. The received signal is passed through a bank of matched filters, one for each waveform, to generate a vector of decision statistics. The matched filter bank is followed by a whitening filter which whitens the noise component of the decision statistics, and creates a causal relation among the decision statistics. The whitened decision statistics are then passed to a modified Field-Based BCJR Algorithm. The modifications to the BCJR algorithm involve the generation of a time-varying trellis, modified branch metrics, and a modified traversal method. Finally the FER performance results for several signal configurations are presented.

A. SIMILARITY TO JOINT MLSD

The signal model and trellis structure for the Joint Field-Based MAP Detector using BCJR is similar in many ways to the Joint MLSD [2] discussed in Chapter II. In both cases a set of matched filters generates decision statistics for each channel such that the decision statistics for channel i are, in general, correlated with the decision statistics for channel j. In the description of the Joint MLSD, based on [73] in Chapter II, a modified branch metric is used, thus eliminating the requirement for a whitening filter. The use of a modified branch metric is known as the Ungerboeck approach [54, 55]. For the Joint Field-based MAP detector it is not immediately obvious how to develop the equivalent modified branch metrics. Note that [104] *does* develop the Ungerboeck model for an ISI channel, but it is still not obvious how to extend this to the time-varying Markov source that results from cochannel reception. For this reason, a whitening filter is used, similar to the Forney approach [51, 55]. The whitening filter has an added benefit of creating a causal discrete-time representation. The output of the whitening filter at sample n is a function of the current input at sample n, and $N_s - 1 + N_c$ previous input samples, where N_s is the number of signals and N_c is the channel memory length, assumed here to be zero. As discussed in Chapter II, the computational complexity of traversing the trellis is exponential with the sample memory, i.e., exponential with $(N_s - 1 + N_c)$.

B. JOINT FIELD-BASED MAP AND BCJR

As in the previous chapter, *a posteriori* probabilities are desired for each possible received field value. Selecting the field value with the maximum *a posteriori* probability minimizes the probability of field error:

$$\hat{\boldsymbol{a}}_k = \arg\max_{\boldsymbol{a_k}}[P(\boldsymbol{a_k}|\boldsymbol{y'})],$$

where $\boldsymbol{a_k}$ represents a field value from the k^{th} signal and $\boldsymbol{y'}$ is the vector of interleaved matched filter outputs. If the source model can be constructed in such a way that it can be written as a Markov chain, then the BCJR algorithm can be used to efficiently perform this calculation.

C. MODEL

Figure 7.1 illustrates the reception of two cochannel signals in AWGN. The symbol vectors $\boldsymbol{\alpha_1}$ and $\boldsymbol{\alpha_2}$, from signal 1 and 2 respectively, are input to pulse shaping filters $\psi_1(t + \tau_1)$ and $\psi_2(t + \tau_2)$. The output of the pulse shaping filters along with AWGN is received at the input to a bank of matched filters. Each filter, h_1 and h_2, is matched to its respective pulse shaping filter ψ_n. The correlated outputs of the two matched filters are

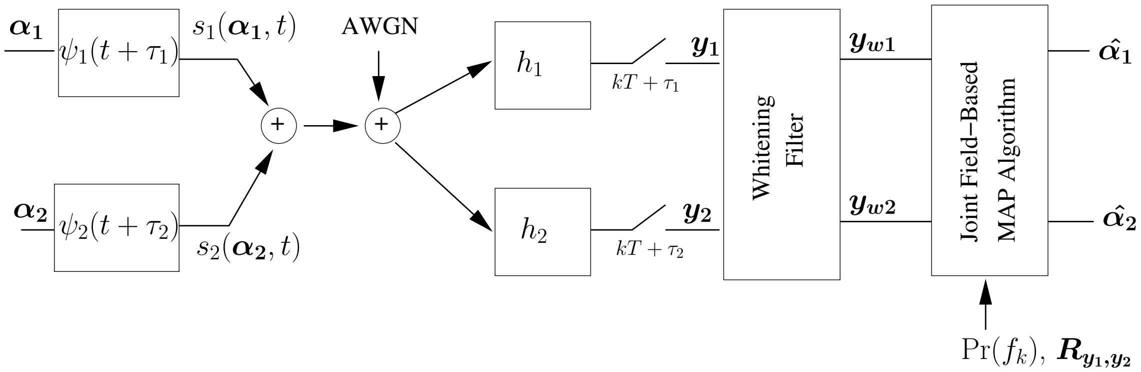


Figure 7.1: Signal Model

whitened with a whitening filter, and the resulting whitened decision statistics are passed to a decision algorithm, the Joint Field-Based MAP Algorithm.

To extend this model to a GMSK signal, as used in AIS, one or more of the Laurent pulses (Appendix A) will be used for ψ. The Laurent pulses have support greater than T_s. This is a relatively straightforward extension but is left out to simplify the development. For an example see [68].

Consider the reception of N_s signals in AWGN:

$$r(t) = \sum_{k=0}^{N_s-1} s_k(t) + z(t); \tag{7.1}$$

where $s_k(t)$ is the complex baseband representation of a signal, and $z(t)$ is a complex AWGN process with power spectral density N_0. Each signal $s_k(t)$ can be represented as a linear combination of amplitude modulated pulses

$$s_k(t) = \sum_i A_k \alpha_k[i] \psi_k(t - iT_s - \tau_k), \tag{7.2}$$

where the unit-energy signal pulse, $\psi_k(t)$, is only nonzero on the interval $[0, T_s]$, and τ_k is the time offset for signal k, where by convention $0 < \tau_j < \tau_k < T_s$, $\forall j < k$. The amplitude of the k^{th} signal is represented by A_k.

The signaling waveforms ψ_k are, in general, not orthogonal to each other. Let ρ describe the inner-product of two synchronous ($\tau_j = \tau_k$) signaling waveforms.

$$\rho = \int_{-\infty}^{\infty} \psi_j(t)\psi_k^*(t)\,\mathrm{d}t \tag{7.3}$$

Let $\rho_{j,k}$ denote the cross-correlation between signaling waveforms corresponding to the correlation associated with symbol n of signal k and symbol n of signal j, where by convention $\tau_j < \tau_k$.

$$\rho_{j,k} = \int_{-\infty}^{\infty} \psi_j(t)\psi_k^*(t - \Delta\tau)\,\mathrm{d}t. \tag{7.4}$$

This corresponds to cross-correlation between the current symbol index of each signal (regions a and d in Figure 7.2).

Let $\rho_{k,j}$ denote the cross-correlation between signaling waveforms corresponding to the correlation associated with symbol n of signal j and symbol $n-1$ of signal k, where by convention $\tau_j < \tau_k$.

$$\rho_{k,j} = \int_{-\infty}^{\infty} \psi_j(t)\psi_k^*(t + (T_s - \Delta\tau))\,\mathrm{d}t \tag{7.5}$$

This corresponds to the cross-correlation between the current and the past (or future) symbol index of each signal (regions c and b in Figure 7.2).

The outputs of the set of matched filters at the sample time are

$$y_k[i] = \int_{-\infty}^{\infty} r(t)\psi_k^*(t - iT_s - \tau_k)\,\mathrm{d}t, \tag{7.6}$$

where ψ_k are the unit-energy[1] impulse responses of the underlying signal waveforms used to generate each signal, s_k. The matched filter outputs $(\boldsymbol{y_0}, \boldsymbol{y_1}, \ldots, \boldsymbol{y_{N_s-1}})$ can be represented as a linear combination of the individual responses from each signal: the desired response, plus the response of $2 \times (N_s - 1)$ symbols from the other channels. This is illustrated in Figure 7.2 for the case of three signals. In the example, the desired response is the second symbol from the second signal $\alpha_2[1]$. The response of the matched filter has a contribution from the previous symbol ($\alpha_3[0]$) and current symbol ($\alpha_3[1]$) from Signal 3, and a contribution from the next symbol ($\alpha_1[2]$) and current symbol ($\alpha_1[1]$) from Signal 1. These contributions are represented by the regions labeled a, b, c, and d in Figure 7.2.

[1] The pulse $\psi_k(t)$ is unit-energy if $\int_{-\infty}^{\infty} \psi_k(t)\psi_k^*(t)\,\mathrm{d}t = 1$

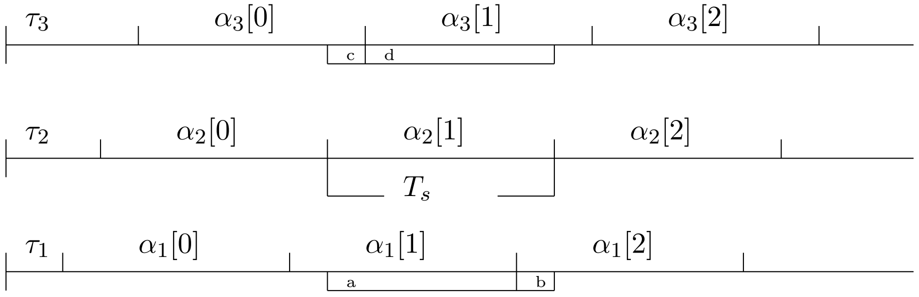


Figure 7.2: Three Asynchronous Cochannel Signals

The contribution of each signal to the output of an individual matched filter is developed below.

Begin by expanding Equation (7.6) using Equation (7.1) and (7.2) to obtain

$$\begin{aligned} y_k[n] &= \int_{-\infty}^{\infty} \underbrace{\left(\sum_{j=1}^{N_s} \underbrace{\sum_i A_j \alpha_j[i] \psi_j(t - iT_s - \tau_j)}_{s_j} + z(t) \right)}_{r(t)} \psi_k^*(t - nT_s - \tau_k)\,\mathrm{d}t \\ &= \sum_{j=1}^{N_s} \sum_i A_j \alpha_j[i] \int_{-\infty}^{\infty} \psi_j(t - iT_s - \tau_j)\psi_k^*(t - nT_s - \tau_k)\,\mathrm{d}t \\ &+ \int_{-\infty}^{\infty} \psi_k^*(t - nT_s - \tau_k) z(t)\,\mathrm{d}t \end{aligned} \tag{7.7}$$

From the finite support of ψ the integral in the double summation of (7.6) can be simplified,

$$\int_{-\infty}^{\infty} \psi_j(t - iT_s - \tau_j)\psi_k^*(t - nT_s - \tau_k)\,\mathrm{d}t = \begin{cases} 0 & \text{if} \quad i < n - 1 \quad \text{or} \quad i > n + 1 \\ 1 & \text{if} \quad j = k, i = n \\ \rho_{j,k} & \text{if} \quad i = n \\ \rho_{k,j} & \text{if} \quad j \neq k, i = n + 1 \\ \rho_{k,j} & \text{if} \quad j \neq k, i = n - 1 \end{cases} \tag{7.8}$$

Then Equation (7.8) can be used to simplify the double summation in Equation (7.7) to obtain

$$\begin{aligned} y_k[n] &= A_k \alpha_k[i] \\ &+ \sum_{j<k} A_j \alpha_j[n] \underbrace{\int_0^{T_s} \psi_j(t) \psi_k^*(t - (\tau_k - \tau_j)) \, \mathrm{d}t}_{\rho_{j,k}} \\ &+ \sum_{j<k} A_j \alpha_j[n+1] \underbrace{\int_0^{T_s} \psi_j(t) \psi_k^*(t - T_s - (\tau_k - \tau_j)) \, \mathrm{d}t}_{\rho_{k,j}} \\ &+ \sum_{j>k} A_j \alpha_j[n-1] \underbrace{\int_0^{T_s} \psi_j(t) \psi_k^*(t + T_s - (\tau_k - \tau_j)) \, \mathrm{d}t}_{\rho_{k,j}} \\ &+ \sum_{j>k} A_j \alpha_j[n] \underbrace{\int_0^{T_s} \psi_j(t) \psi_k^*(t - (\tau_k - \tau_j)) \, \mathrm{d}t}_{\rho_{j,k}} \\ &+ z_k[i] \end{aligned} \tag{7.9}$$

where $z_k[i]$ is a complex Gaussian random variable with variance $\sigma^2 = N_0$. The Gaussian random process formed by interleaving the noise components from the output of each matched filter is given by $\boldsymbol{z}[i] = \begin{bmatrix} z_1[i] & z_2[i] & z_3[i] & \ldots & z_{N_s}[i] \end{bmatrix}^T$, and has autocorrelation matrices [2]

$$\mathcal{E}\{\boldsymbol{z}[i]\boldsymbol{z}^{*T}[j]\} = \begin{cases} \sigma^2 \boldsymbol{R}^T[1], & \text{if} j = i+1 \\ \sigma^2 \boldsymbol{R}^T[0], & \text{if} j = i \\ \sigma^2 \boldsymbol{R}^*[1], & \text{if} j = i-1 \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{7.10}$$

where $\boldsymbol{R}[i-j]$ is the $(N_s \times N_s)$ matrix with elements n, m given by [2]

$$R_{n,m}[i-j] = \int_{-\infty}^{\infty} \psi_n(t - iT_s - \tau_n) \psi_m^*(t - jT_s - \tau_m) \, \mathrm{d}t \tag{7.11}$$

D. WHITENED MATCHED FILTER

To easily calculate the branch metrics, and to reduce the number of symbols upon which a decision statistic depends, the decisions statistics are passed through a whitening filter. Take the N_s length-N vectors of matched filter outputs and interleave them to form a vector of length-(N_sN) matched filter outputs

$$\boldsymbol{y'} = \Big[y_0[0] \quad y_1[0] \quad \ldots \quad y_{N_s-1}[0] \quad y_0[1] \quad y_1[1] \quad \ldots \quad y_0[N-1] \quad y_1[N-1] \ldots y_{N_s-1}[N-1]\Big]^T \tag{7.12}$$

The idea of the whitening filter is to perform a linear operation on the interleaved vector $\boldsymbol{y'}$ that removes the correlation between the noise components.

The interleaved vector of matched filter outputs is written in matrix form as

$$\boldsymbol{y'} = \boldsymbol{RA\alpha'} + \boldsymbol{z'}, \tag{7.13}$$

where $\boldsymbol{A}$ is the diagonal matrix of signal amplitudes. The vector $\boldsymbol{\alpha'}$ is the interleaved vector of length N transmitted symbol vectors

$$\boldsymbol{\alpha'} = \Big[\alpha_0[0] \quad \alpha_1[0] \quad \ldots \quad \alpha_{N_s-1}[0] \quad \alpha_0[1] \quad \alpha_1[1] \quad \ldots \quad \alpha_0[N-1] \quad \alpha_1[N-1] \ldots \alpha_{N_s-1}[N-1]\Big]^T \tag{7.14}$$

and $\boldsymbol{z'}$ is the vector of interleaved correlated noise components of the decision statistics $\boldsymbol{y'}$, with $\mathcal{E}\{\boldsymbol{z'}\boldsymbol{z'}^H\} = \sigma^2\boldsymbol{R}$. The matrix $\boldsymbol{R}$ is the global $N_sN \times N_sN$ correlation matrix for the noise components of the matched filter outputs

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}[0] & (\boldsymbol{R}[1])^T & 0 & 0 & \ldots & 0 \\ \boldsymbol{R}[1] & \boldsymbol{R}[0] & (\boldsymbol{R}[1])^T & 0 & \ldots & 0 \\ 0 & \boldsymbol{R}[1] & \boldsymbol{R}[0] & (\boldsymbol{R}[1])^T & \ldots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ldots & \ddots \\ 0 & 0 & 0 & 0 & \boldsymbol{R}[1] & \boldsymbol{R}[0] \end{bmatrix} \tag{7.15}$$

where the elements of the matrices $\boldsymbol{R}[0]$ and $\boldsymbol{R}[1]$ are defined in Equation 7.11.

For illustration, a 4 × 4 correlation matrix for two signals is given below in Equation (7.16)

$$\boldsymbol{R} = \begin{bmatrix} 1 & \rho_{12} & 0 & 0 \\ \rho_{12} & 1 & \rho_{21} & 0 \\ 0 & \rho_{21} & 1 & \rho_{12} \\ 0 & 0 & \rho_{12} & 1 \end{bmatrix} \tag{7.16}$$

where ρ_{12} and ρ_{21} are the cross-correlation coefficients defined in Equation (7.4) and (7.5).

Because the correlation matrix $\boldsymbol{R}$ is positive definite, the Cholesky decomposition of $\boldsymbol{R}$ can be used to find an upper triangular matrix $\boldsymbol{U}$, such that $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{U}^T$ [103, 105]. The noise components of $\boldsymbol{y'}$ are then whitened by applying the transformation $\boldsymbol{U}^{-1}$ to the interleaved vector of decision statistics defined in Equation (7.13). This transformation results in a new whitened set of decision statistics $\boldsymbol{y'_w}$ [106]

$$\begin{aligned} \boldsymbol{y'_w} &= \boldsymbol{U}^{-1}\boldsymbol{y'} \\ &= \boldsymbol{U}^{-1}\boldsymbol{R}\boldsymbol{A}\boldsymbol{\alpha'} + \boldsymbol{U}^{-1}\boldsymbol{z'} \\ &= \boldsymbol{U}^{-1}\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{\alpha'} + \boldsymbol{U}^{-1}\boldsymbol{z'} \\ &= \boldsymbol{U}^T\boldsymbol{A}\boldsymbol{\alpha'} + \boldsymbol{z'_w} \end{aligned} \tag{7.17}$$

where $\boldsymbol{z'_w}$ is the whitened version of $\boldsymbol{z'}$ with $\mathcal{E}\{\boldsymbol{z'_w}\boldsymbol{z'_w}^H\} = \sigma^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. Since the matrix $\boldsymbol{U}^{-1}$ is also an upper triangular matrix [107]; the whitening transform can be thought of as a noncausal filtering operation. This transformation not only whitens the noise components of the matched filter outputs, but also generates a new vector of sufficient statistics $\boldsymbol{y'_w}$ whose signal components are a function of the current and past $N_s - 1$ symbols. The causal transform results because the matrix $\boldsymbol{U}^T$ is lower triangular, so the resulting vector is a function of current and past values rather than current, past and future values. The vector will only depend on $N_s - 1$ past values, rather than all past values, because $\boldsymbol{U}^T$ is banded[2]. This is in contrast to the vector $\boldsymbol{y'}$ whose elements are also a function of the future $N_s - 1$ symbols.

This transformation is a time-varying finite impulse response (FIR) filter on the matched filter outputs $\boldsymbol{y'}$. The transformation is time-varying because the transformation

[2] An $n \times n$ matrix $\boldsymbol{A}$ with elements $a_{i,j}$ is banded if $a_{i,j} = 0 \quad \forall j < i - k_1$ and $a_{i,j} = 0 \quad \forall j > i + k_2$ for some $0 < k_1, k_2 < n$

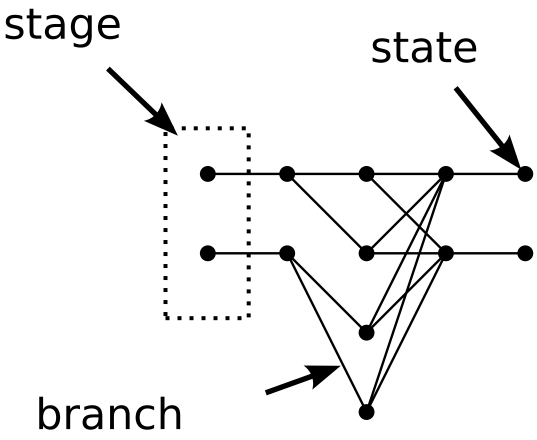


Figure 7.3: Nomenclature for Trellis

matrix $\boldsymbol{U}^{-1}$ is, in general, not Toeplitz. The outputs of the whitened matched filter, $\boldsymbol{y_w'}$, may now be modeled as a Markov chain received through an AWGN channel.

E. GENERATION OF THE TRELLIS

The noiseless portion of $\boldsymbol{y_w'}$ can be modeled as a time-varying Markov chain. The trellis is a tool used to represent the Markov chain as time progresses. The trellis consists of $N+1$ stages, where N is the length of the whitened interleaved decision statistics, $\boldsymbol{y_w'}$. Each stage consists of a number of states representing the permutations of the memory of past transmitted symbols. The memory of each stage is large enough to account for the ISI or signal overlap, and also to account for the *a priori* probabilities for the fields. For a binary alphabet, each state in each stage has no more than two branches connecting it to a state in the next stage: one branch for each of the two possible symbol values. Each state in each stage will then have one or more branches connecting to a state from the previous stage. This is illustrated in Figure 7.3.

The states in the trellis are defined in such a way that the Markov property of the state transitions is preserved

$$\Pr[\lambda[n+1] = q|\lambda[n] = p] = \Pr[\lambda[n+1] = q|\lambda[n] = p_n, \lambda[n-1] = p_{n-1}, \ldots, \lambda[0] = p_0]. \quad (7.18)$$

For field widths greater than one symbol, this results in a time-varying trellis; this is illustrated below.

Table 7.1: Signal 0 Structure

Field Number	Width	Values / Prior Pairs		
0	1	(0, 0.5)	(1, 0.5)	
1	3	(000, 0.2)	(101, 0.6)	(111, 0.2)
2	1	(0, 0.5)	(1, 0.5)	
3	1	(0, 0.5)	(1, 0.5)	

Table 7.2: Signal 1 Structure

Field Number	Width	Values / Prior Pairs	
0	1	(1, 0.5)	(0, 0.5)
1	2	(01, 0.5)	(00, 0.5)
2	1	(0, 0.5)	(1, 0.5)
3	2	(00, 0.5)	(11, 0.5)

Consider two signals with parameters listed in Table 7.1 and Table 7.2. Each signal contains a various number of fields; each field with a various number of symbols. For each field there is a set of possible values and associated prior probabilities p_k. The signals are received at a single receiver with their respective time offsets, τ_0 and τ_1, with $\tau_0 < \tau_1$. After the whitening filter, outputs at time n is a function of the previous $N_s - 1$ symbols. The time-varying trellis is generated in such a way that the Markov property (7.18) is maintained. This property is maintained if all the overlapping fields of the current symbol are stored in the current state.

1. Determining the Stage Memory

The formation of the trellis begins by determining the required number of past symbols represented by a stage (the memory of the stage). This involves looking back from the current symbol at all of the overlapping fields. The basic algorithm is outlined in Table 7.4. The algorithm starts with the interleaved vector of whitened matched filter outputs $\boldsymbol{y}'_{\boldsymbol{w}}$. Each value in $\boldsymbol{y}'_{\boldsymbol{w}}$ is associated with a transition to a new stage in the trellis. Each value in $\boldsymbol{y}'_{\boldsymbol{w}}$ is also associated with a specific field of a specific signal. The first symbol included in the memory for a stage is the first symbol from the set of overlapping fields. The last symbol to be represented by a stage is the symbol associated with the current element of $\boldsymbol{y}'_{\boldsymbol{w}}$. This results in the M symbols represented by this stage. Next, the number of unique elements represented by the M symbols must be determined; this is the number of states.

As an example of determining the memory of a stage, consider Figure 7.4 based on the signals listed in Tables 7.1 and 7.2.

- At Stage 4, corresponding to the reception of symbol $a_1[1]$ on the Stage 3 to Stage 4 transition, the current field ($f_1[1]$) does not overlap the first symbol in Stage 3 memory ($a_1[0]$), but the next field ($f_0[1]$) does. Therefore, the first element represented by this stage remains the same, and the states will represent combinations of the following symbols: $a_1[0], a_0[1], a_1[1]$.

- At Stage 5, corresponding to the reception of symbol $a_0[2]$ on the Stage 4 to Stage 5 transition, the current field ($f_0[1]$) overlaps the first symbol in Stage 4 memory ($a_1[0]$). Therefore, the first element represented by the stage remains the same. The states will represent combinations of the following symbols: $a_1[0], a_0[1], a_1[1], a_0[2]$.

- At Stage 6, corresponding to the reception of symbol $a_1[2]$ on the Stage 5 to Stage 6 transition, field $f_0[1]$ is still not complete. Therefore, the states will represent combinations of the following symbols: $a_1[0], a_0[1], a_1[1], a_0[2], a_1[2]$.

- At Stage 7, corresponding to the reception of symbol $a_0[3]$ on the Stage 6 to Stage 7 transition, the current field ($f_0[1]$) does not continue, and the next field ($f_1[2]$) is not a continuation. Therefore, only $a_0[3]$ is required in memory; the states will represent combinations of the following symbol $a_0[3]$.

Table 7.3 lists the symbols comprising the memory of each stage based on the two signals listed in Tables 7.1 and 7.2.

2. Determining the Number of States per Stage

As discussed in Chapter VI, some field values have negligible *a priori* probability. Ignoring field values with negligible probability leads to a reduction in the number of states required when constructing the trellis. An algorithm for determining the number of states is shown in Table 7.6. The algorithm starts by determining the number of elements as outlined in Table 7.4. Continuing with the example above, observe from Table 7.3 that the values stored in a state can span multiple fields. To find the number of states, generate all possible partial field values based on the symbols of a field represented by the memory of the current stage. Finally, find the unique words after this masking, and assign a state for each unique word.

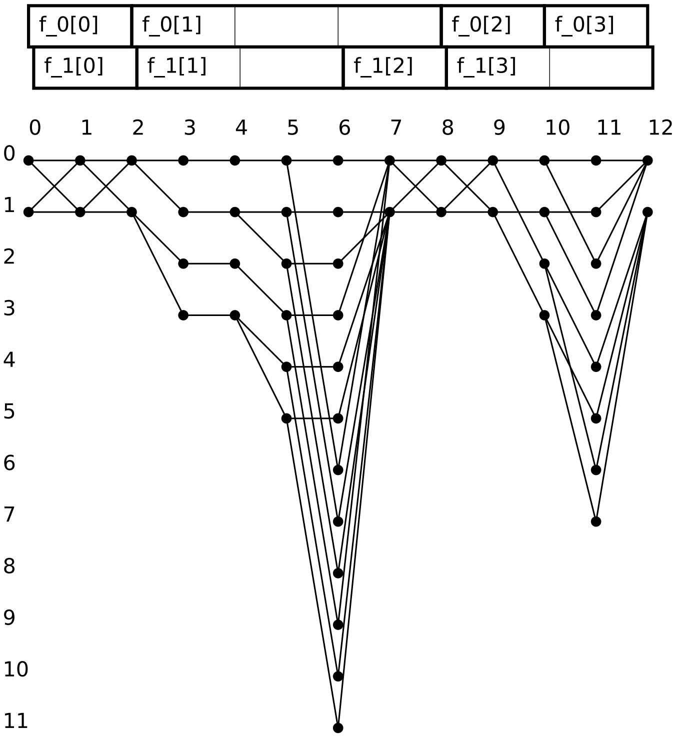


Figure 7.4: Example Trellis Using Fields Described in Tables 7.1 and 7.2

Table 7.3: Number Of Elements Per Stage

Stage Number	Elements				
1	$a_0[0]$				
2	$a_1[0]$				
3	$a_1[0]$	$a_0[1]$			
4	$a_1[0]$	$a_0[1]$	$a_1[1]$		
5	$a_1[0]$	$a_0[1]$	$a_1[1]$	$a_0[2]$	
6	$a_1[0]$	$a_0[1]$	$a_1[1]$	$a_0[2]$	$a_1[2]$
7	$a_0[3]$				
8	$a_1[3]$				
9	$a_0[4]$				
10	$a_0[4]$	$a_1[4]$			
11	$a_0[4]$	$a_1[4]$	$a_0[5]$		
12	$a_1[5]$				

Table 7.4: Determining the number of elements represented by a stage

```
for all Elements of $\boldsymbol{y'_w}$ do
  CurrentFirstElement $\Leftarrow$ Current Index
  OtherFirstElement $\Leftarrow$ Current Index
  if Current Field Continuing then
    CurrentFirstElement $\Leftarrow$ Beginning of current field $-1$
  end if
  if Other Field Continuing then
    OtherFirstElement $\Leftarrow$ Beginning of other field $-1$
  end if
  FirstElement $\Leftarrow$ min(OtherFirstElement,CurrentFirstElement)
end for
```

Table 7.5: Permutations of Field Values for Stage 4

$f_1[0]$	$f_1[1]$	$f_0[1]$
$\underline{1}$	$\underline{0}0$	$\underline{0}00$
$\underline{0}$	$\underline{0}0$	$\underline{0}00$
$\underline{1}$	$\underline{0}1$	$\underline{0}00$
$\underline{0}$	$\underline{0}1$	$\underline{0}00$
$\underline{1}$	$\underline{0}0$	$\underline{1}01$
$\underline{0}$	$\underline{0}0$	$\underline{1}01$
$\underline{1}$	$\underline{0}1$	$\underline{1}01$
$\underline{0}$	$\underline{0}1$	$\underline{1}01$
$\underline{1}$	$\underline{0}0$	$\underline{1}11$
$\underline{0}$	$\underline{0}0$	$\underline{1}11$
$\underline{1}$	$\underline{0}1$	$\underline{1}11$
$\underline{0}$	$\underline{0}1$	$\underline{1}11$

For example, consider Stage 4 of Figure 7.4 representing the following symbols $a_1[0], a_0[1], a_1[1]$, this stage contains symbols from fields zero and one from Signal 1 ($f_1[0]$, $f_1[1]$). Stage 4 also contains symbols from Field 1 of Signal 0 ($f_0[1]$). The possible permutations of these three fields with the symbols $a_1[0], a_0[1], a_1[1]$ highlighted, are shown in Table 7.5 Although there are 12 permutations of field values, Stage 4 only requires four states; the other permutations of the fields are for symbols that are not contained the Stage 4 memory.

Table 7.6: Determining the number of required states for each stage

```
**for all** Stages **do**
  Determine number of elements for this stage
  Determine the number of unique groups in this set of elements
  **for all** Unique Element of Set **do**
    Assign State Element of Set
  **end for**
**end for**
```

3. Making the Branch Connections

Once the values represented by each state are determined, the next step is to make the branch connections. Table 7.7 summarizes the algorithm for connecting the states from one stage to the next stage. At each stage this involves a loop through all of the states at the next stage for each state at the current stage. Working from left to right across the trellis, at Stage n State p there are at most K branches leaving each state, where K is the size of the alphabet for the symbols. A particular state represents a permutation of the received symbols represented by this stage. A branch exists from State p in Stage n to State q in Stage $n + 1$ if the mutual symbols in both states are all the same value. If a particular value for the next symbol does not lead to a field value with a probability of occurring greater than ϵ, $(p_k > \epsilon)$, then the state transition will not be drawn. The resulting trellis with branch connections is shown in Figure 7.5. Consider the branches connecting Stage 8 to Stage 9 in Figure 7.5. There is only one branch leaving State 0. This is to be interpreted as follows: with the information in State 0 at Stage 8 there is only one possible value for the next symbol.

F. BRANCH METRIC CALCULATION

The branch metric calculation for the field-based BCJR algorithm requires a slight modification from the standard method outlined in Appendix B. The term γ represents the probability of transition from State p at Stage n to State q at Stage $n + 1$ with the

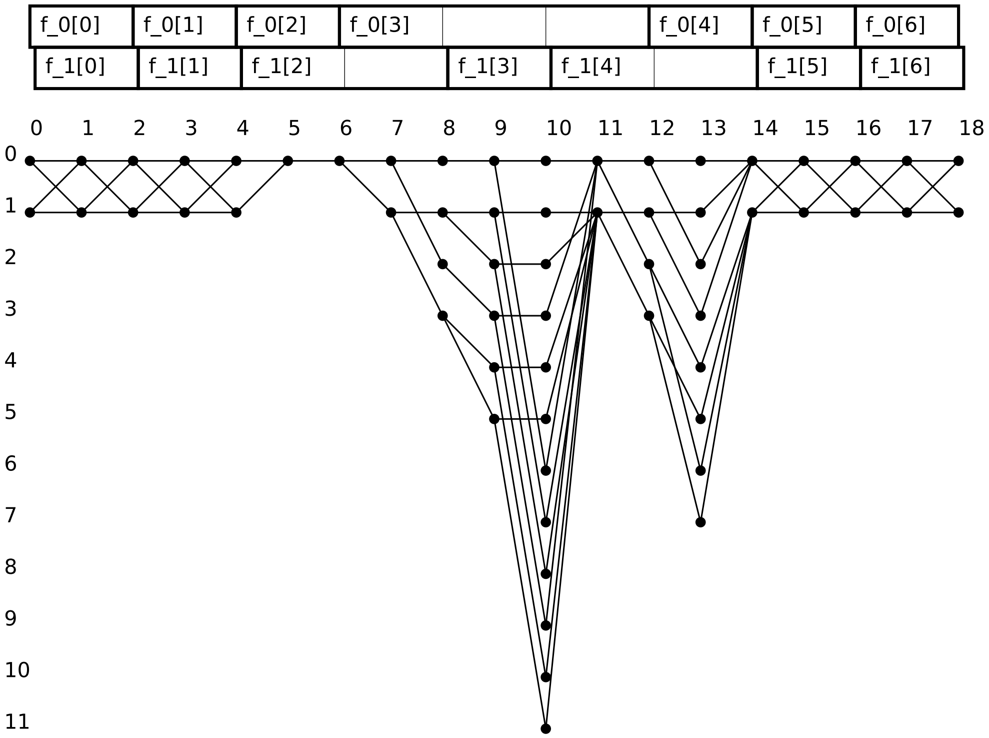


Figure 7.5: Resulting Joint MAP Trellis

Table 7.7: Determining the Branch Connections

```
for all stage do
  for all cstate in current stage do
    shift current values over one bit
    mask off overlapping elements
    for all nstate in next stage do
      if cstate in current stage contains same elements nstate then
        add branch connection
      end if
    end for
  end for
end for
```

observation y_n. The term γ can be factored as follows:

$$\begin{aligned}\gamma_n(p,q) &= \Pr(\lambda[n+1]=q, y_n|\lambda[n]=p) \\ &= \Pr(y_n|\lambda[n+1]=q, \lambda[n]=p)\underbrace{\Pr(\lambda[n+1]=q|\lambda[n]=p)}_{\textit{a priori} \text{ probabilities}}\end{aligned} \tag{7.19}$$

where $\lambda[n]$ is the state at Stage n. The second factor is the *a priori* probability $\Pr(a_x = a)$. The first factor can be rewritten as $\Pr(y_n|s_{p,q})$ where $s_{p,q}$ is the output of the channel uniquely associated with a state transition from p to q. Because the noise components have been whitened, the conditional probability can be written as

$$\Pr(y_n|s_{p,q}) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(\frac{-\|y_n - s_{p,q}\|^2}{2\sigma^2}\right), \tag{7.20}$$

where σ^2 is the variance of the noise term from Equation 7.17 and $\|(\cdot)\|^2$ is the L2-norm.

The modification for field-based BCJR is in the second factor, the *a priori* probability $\Pr(a_x = a)$. The algorithm is provided *a priori* field probabilities; a specific field probability is represented by the transition from stage to stage on the last bit of a field. If the state transition does not represent the last symbol in a field, then this probability is set to one. The *a priori* probability for the branch metric could arbitrarily be set to any constant value for each transition; the effect is removed during normalization. If the state transition does represent the last symbol in a field, then use the *a priori* probability for that field value [108].

G. TRELLIS TRAVERSAL

Trellis traversal is identical to the standard BCJR algorithm detailed in Appendix B. For the forward pass, the values for α at Stage n and State q are calculated by the following recursive equation:

$$\begin{aligned}\alpha_{n+1}(q) &= \sum_{p=0}^{Q-1} \underbrace{\Pr(\lambda[n]=p, \boldsymbol{y}_{<n})}_{\alpha_n(p)} \underbrace{\Pr(\lambda[n+1]=q, y_n|\lambda[n]=p)}_{\gamma_n(p,q)} \\ &= \sum_{p=0}^{Q-1} \alpha_n(p)\gamma_n(p,q)\end{aligned} \tag{7.21}$$

where Q is the number of states at Stage n.

In a similar manner for the backwards pass, the βs at Stage n and State p are given by the following recursive equation:

$$\begin{aligned}\beta_n(p) &= \sum_{q=0}^{Q-1} \underbrace{\Pr(\lambda[n+1]=q, y_n|\lambda[n]=p)}_{\gamma_n(p,q)} \underbrace{\Pr(\boldsymbol{y}_{>n}|\lambda[n+1]=q)}_{\beta_{n+1}(q)} \\ &= \sum_{q=0}^{Q-1} \gamma_n(p,q)\beta_{n+1}(q)\end{aligned} \tag{7.22}$$

where Q is the number of states in the next state at time $n+1$.

After calculating the αs and the βs, the *a posteriori* probabilities are then given by

$$\begin{aligned}\Pr(a=a_x|\boldsymbol{y}) &= \frac{\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})}{\sum_{a_x\in\mathcal{A}}\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})} \\ &= \frac{\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)}{\sum_{a_x\in\mathcal{A}}\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)}\end{aligned} \tag{7.23}$$

where $\mathcal{S}_x$ is the set of transitions associated with the specific input a_x and $a_x \in \mathcal{A}$ where $\mathcal{A}$ is a finite alphabet.

The effort to preserve the Markov property is only to keep track of the *a priori* field probabilities. For example, the received values in $\boldsymbol{y}'_{\boldsymbol{w}}$ used in the metric for the values for γ (branch metrics) only depend on the last $N_s - 1$ symbols placed into the state and the current symbol. This is the memory from the channel, or equivalently from the asynchronous overlapping signals. The additional state memory beyond $N_s - 1$ allows for book keeping of the *a priori* probabilities. A consequence of this is that many of the branch metrics (γ) are identical; they are not be a function of the additional stage memory.

H. RESULTS

This section presents the FER performance results for the Joint Field-Based MAP Detector. Performance results of the detector are presented for two signals with a simple structure leading to the Joint MLSD developed in Chapter II. Results are then presented for several other more complex signal configurations. Table 7.8 summarizes the cases investigated in this section.

Table 7.8: Table Summarizing Test Cases

Case	Description
A	Simple case, equally likely independent symbols, each field is length one
D	3-symbol fields with two equally likely values 0 and 1
F	3-symbol fields with general distributions for field values
B	various field widths, various field value distributions

1. Case A:

In this first example the performance of the Joint Field-Based MAP Detector is demonstrated for the structure of the signals such that it reduces to a joint MLSD.

The structure of the two signals are shown in Table 7.9 and Table 7.10. Each signal consists of single-symbol fields which take on one of two values with equal *a priori* probability. Notice that the structure of the trellis in Figure 7.6 is identical to the structure of the joint MLSD introduced in Chapter II. This simple structure results for the two signal case when the fields are only one symbol wide.

a. ***Signal Offset*** $\Delta\tau = 0T_s$

First, it is necessary to point out that here $\Delta\tau = 0T_s$, and there is no ISI introduced by the transmitter or channel, therefore there is no need for the trellis approach. The optimum receiver can be developed as in Chapter VI. The example of $\Delta\tau = 0T_s$ is used to demonstrate that for this case the algorithm performs equivalently to the method outlined in Chapter VI. In this first example FER curves are plotted (Figure 7.7) for field three of single one with various values for the cross-correlation coefficient ρ defined in Equation (7.3).

Table 7.9: Case A: Signal 0 Structure

Field Number	Width	Values / Prior Pairs	
0	1	(0, 0.5)	(1, 0.5)
1	1	(1, 0.5)	(0, 0.5)
2	1	(1, 0.5)	(0, 0.5)
3	1	(1, 0.5)	(0, 0.5)
4	1	(1, 0.5)	(0, 0.5)

Table 7.10: Case A: Signal 1 Structure

Field Number	Width	Values / Prior Pairs	
0	1	(1, 0.5)	(0, 0.5)
1	1	(1, 0.5)	(0, 0.5)
2	1	(1, 0.5)	(0, 0.5)
3	1	(1, 0.5)	(0, 0.5)
4	1	(1, 0.5)	(0, 0.5)

Table 7.11: Case A: Number Of Elements Per Stage

Stage Number	Elements
1	$a_0[0]$
2	$a_1[0]$
3	$a_0[1]$
4	$a_1[1]$
5	$a_0[2]$
6	$a_1[2]$
7	$a_0[3]$
8	$a_1[3]$
9	$a_0[4]$
10	$a_1[4]$

Notice that with $\rho = 1$ the FER is approximately 0.5 (actually it approaches 0.25 for high SNR, of the four possible transmitted pairs only (-1,$+1$) and ($+1$,-1) lead to an ambiguous result that will likely result in an error). Notice that the BER is identical to the results presented in Chapter II for Joint Detection. For a cross-correlation coefficient of $\rho = 0$, the performance is identical to single-channel binary antipodal signaling. For small cross-correlation coefficient, the FER performance is very close to binary antipodal signaling.

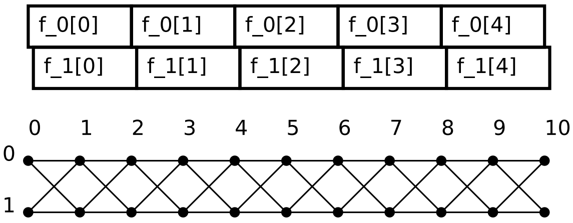


Figure 7.6: Case A: Trellis

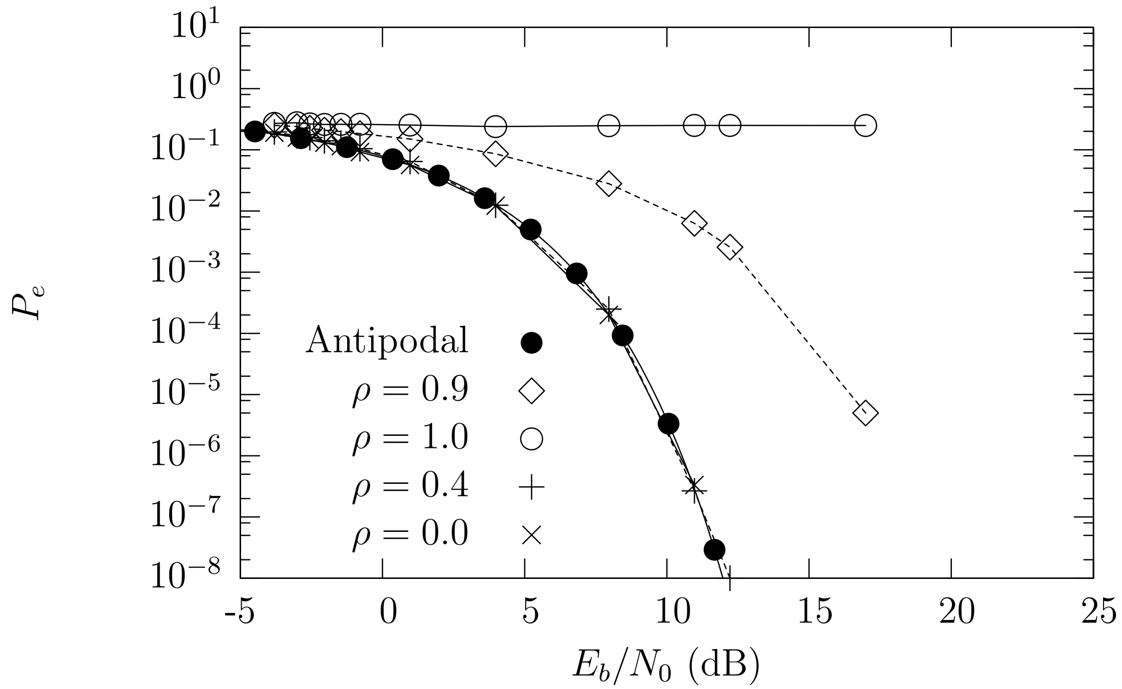


Figure 7.7: Case A: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, synchronous case $\Delta\tau = 0T_s$

b. Signal Offset $\Delta\tau = 0.5T_s$

With an offset between the two signals reasonable FER performance can be achieved even with highly correlated signals. This is similar to the example outlined in Section I.E. Although the two fundamental pulse shapes have a cross-correlation coefficient of $\rho = 1$, the time offset leads to $\rho_{12} = \rho_{21} = 0.5$. Because of this, the signals can be resolved. Figure 7.8 illustrates this for the third field of signal one; even for a cross-correlation coefficient of 1.0 the FER performance is acceptable.

There is a variation in performance based on the distances from the start of the packet; in order to make the decision on symbol n, all symbols received prior to n must not be in error (or there must be an even number of errors). Figure 7.9 presents FER performance for various field numbers; larger field numbers correspond to greater distances from the start of a packet. Notice that the error rate is higher for fields that are further away from the beginning of the packet.

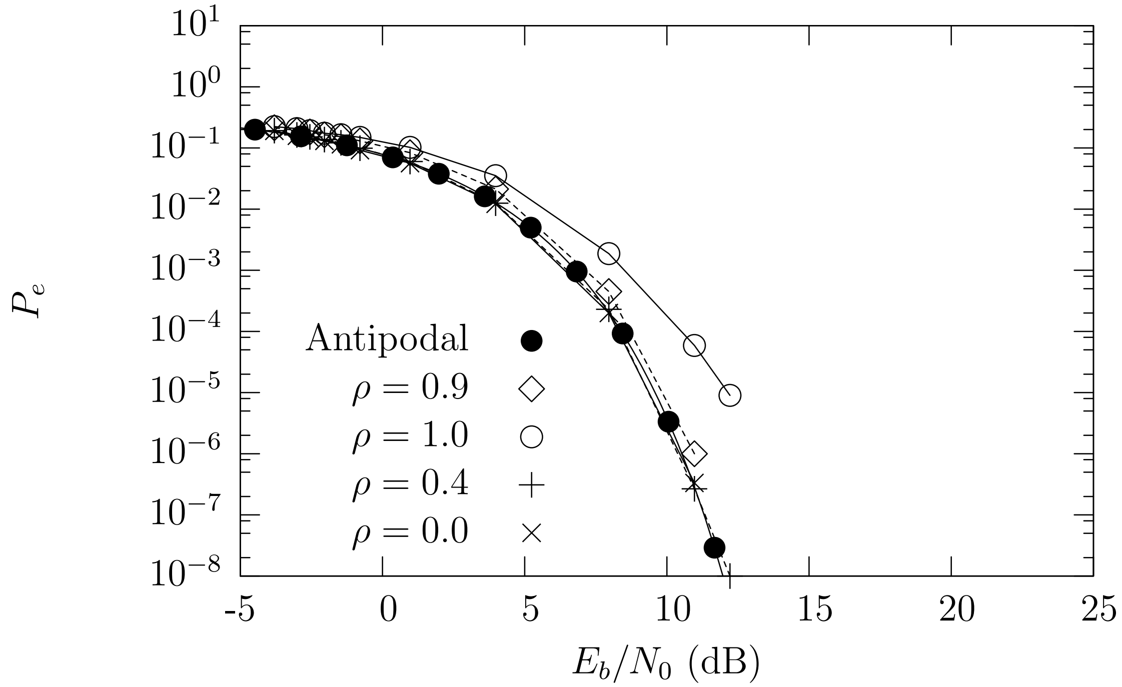


Figure 7.8: Case A: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, asynchronous case $\Delta\tau = 0.5T_s$

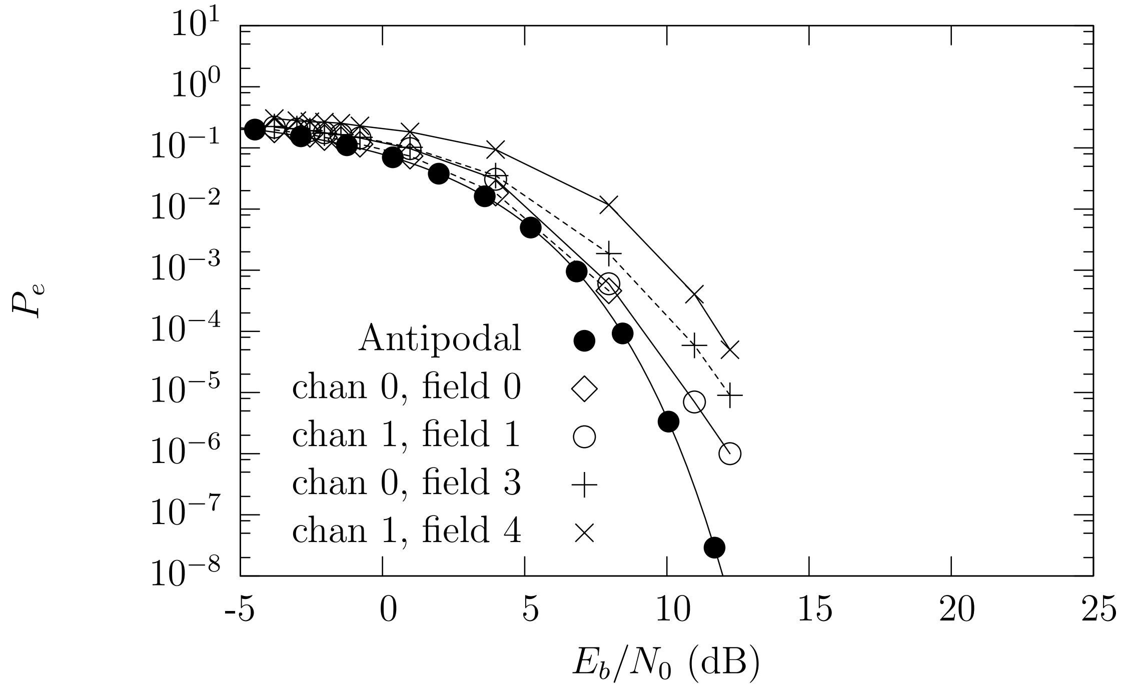


Figure 7.9: Joint Field-Based MAP Performance Curves Variation Based on Distance from Start of Packet Correlation Coefficient $\rho = 1$, asynchronous case $\Delta\tau = 0.5T_s$

2. Case D:

In this example, the FER performance of the Joint Field-Based MAP Detector for signals with a fixed field width of three symbols is examined. Each field takes on one of two possible values, 000 or 001. We might expect the performance in this example to be very similar to that of the previous section,(VII.H.1); as illustrated below, this is not the case. The structure of the signals is shown in Tables 7.12 and 7.13. The structure of the resulting trellis is illustrated in Figure 7.10. Notice the simple state transitions that result from the signal structure.

Table 7.12: Case D: Signal 0

Field Number	Width	Values / Prior Pairs	
0	3	(001, 0.5)	(000, 0.5)
1	3	(001, 0.5)	(000, 0.5)
2	3	(001, 0.5)	(000, 0.5)
3	3	(001, 0.5)	(000, 0.5)
4	1	(0, 1)	

Table 7.13: Case D: Signal 1

Field Number	Width	Values / Prior Pairs	
0	3	(001, 0.5)	(000, 0.5)
1	3	(001, 0.5)	(000, 0.5)
2	3	(001, 0.5)	(000, 0.5)
3	3	(001, 0.5)	(000, 0.5)
4	1	(0, 1)	

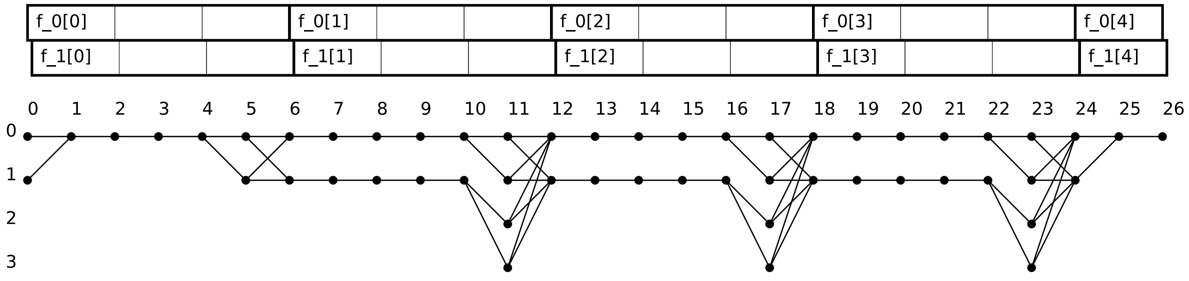


Figure 7.10: Case D: Trellis

Table 7.14: Case D: Number Of Elements Per Stage

Stage Number	Elements					
1	$a_0[0]$					
2	$a_0[0]$	$a_1[0]$				
3	$a_0[0]$	$a_1[0]$	$a_0[1]$			
4	$a_0[0]$	$a_1[0]$	$a_0[1]$	$a_1[1]$		
5	$a_0[0]$	$a_1[0]$	$a_0[1]$	$a_1[1]$	$a_0[2]$	
6	$a_1[2]$					
7	$a_1[2]$	$a_0[3]$				
8	$a_1[2]$	$a_0[3]$	$a_1[3]$			
9	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$		
10	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	$a_1[4]$	
11	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	$a_1[4]$	$a_0[5]$
12	$a_1[5]$					
13	$a_1[5]$	$a_0[6]$				
14	$a_1[5]$	$a_0[6]$	$a_1[6]$			
15	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$		
16	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$	$a_1[7]$	
17	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$	$a_1[7]$	$a_0[8]$
18	$a_1[8]$					
19	$a_1[8]$	$a_0[9]$				
20	$a_1[8]$	$a_0[9]$	$a_1[9]$			
21	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$		
22	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$	$a_1[10]$	
23	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$	$a_1[10]$	$a_0[11]$
24	$a_1[11]$					
25	$a_0[12]$					
26	$a_1[12]$					

a. Signal Offset $\Delta\tau = 0T_s$

Figure 7.11 presents the FER performance of the Joint Field-Based MAP Detector for synchronous signals and various cross-correlation coefficients. The performance here is identical to the results for Case A presented in Section VII.H.1.

b. Signal Offset $\Delta\tau = 0.5T_s$

Figure 7.12 presents the FER performance for Joint Field-Based MAP Detector for the asynchronous case for various cross-correlation coefficients. The performance for this signal structure is superior to the results presented for Case A in Section VII.H.1.

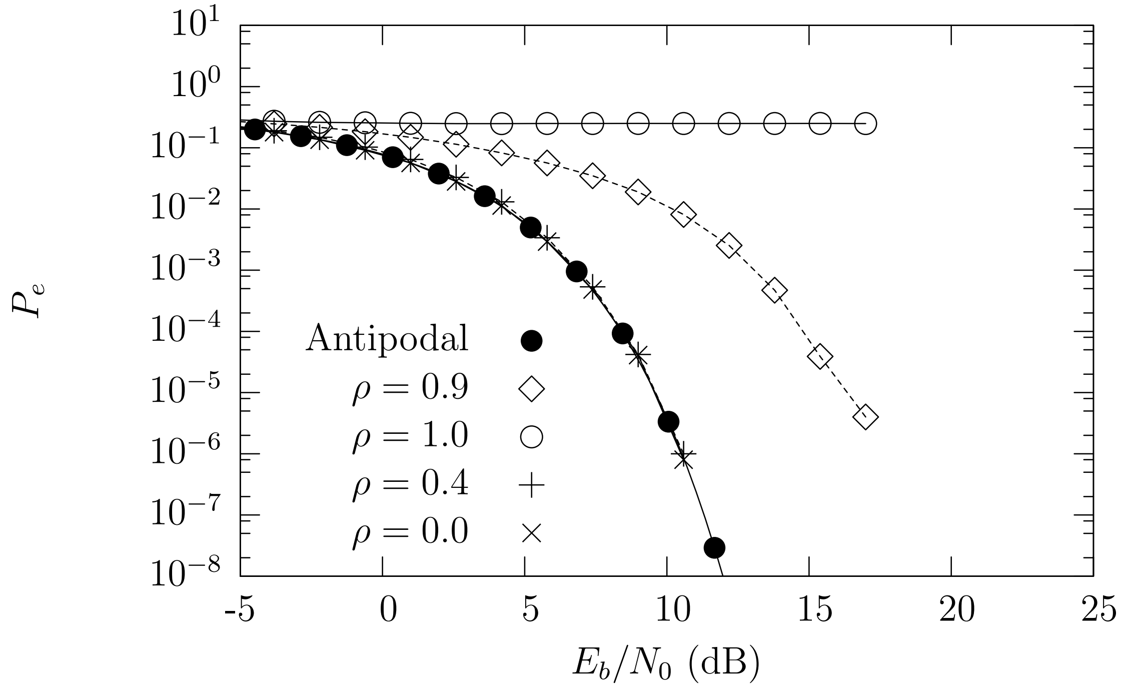


Figure 7.11: Case D: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, synchronous case $\Delta\tau = 0T_s$, fixed width fields

The performance is only a fraction of one dB from the antipodal signaling case. The reason the detector performs so well, even with $\rho = 1$, is that each of the three bit fields only has one unknown bit. With an offset between the two signals, $\Delta\tau > 0$, the unknown bits in each field partially overlap the known bits in the other signal.

In Figure 7.13 the FER performance for the Joint Field-Based MAP Detector are presented for the asynchronous case for various Fields. The dependence of error rate on distance from the beginning of a packet, in contrast with Case A in the previous section, is not seen here. For Case A there is a requirement that the previous n fields not be in error; in this case, because of the known symbols within the fields, there is no such requirement.

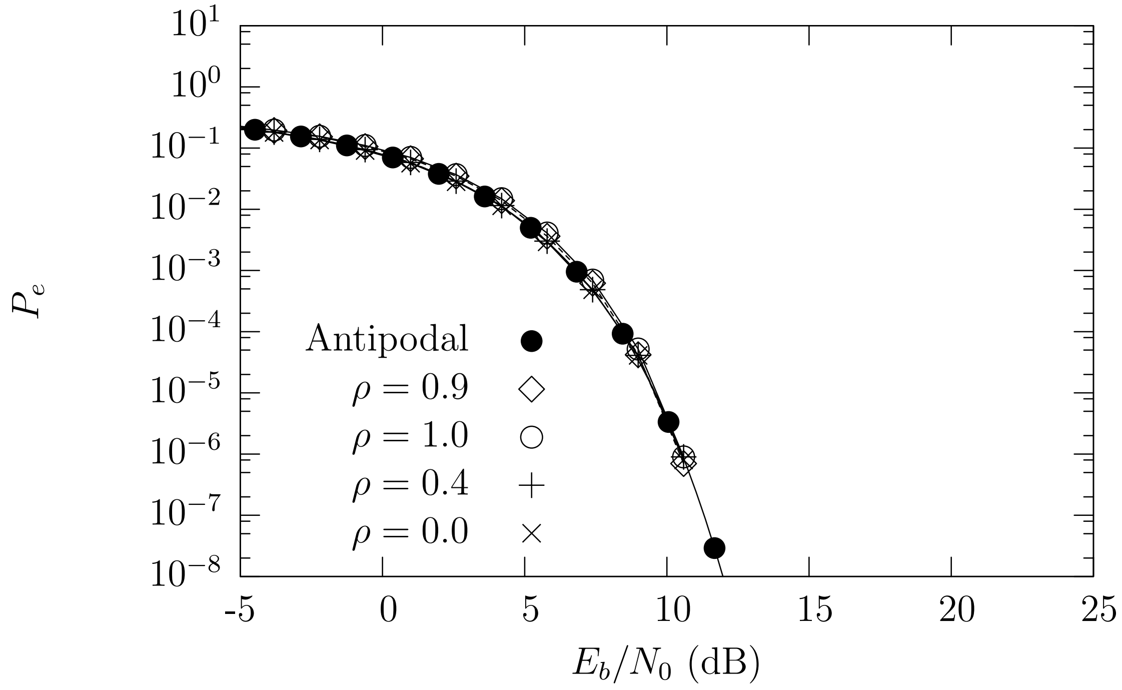


Figure 7.12: Case D: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, asynchronous case $\Delta\tau = 0.5T_s$

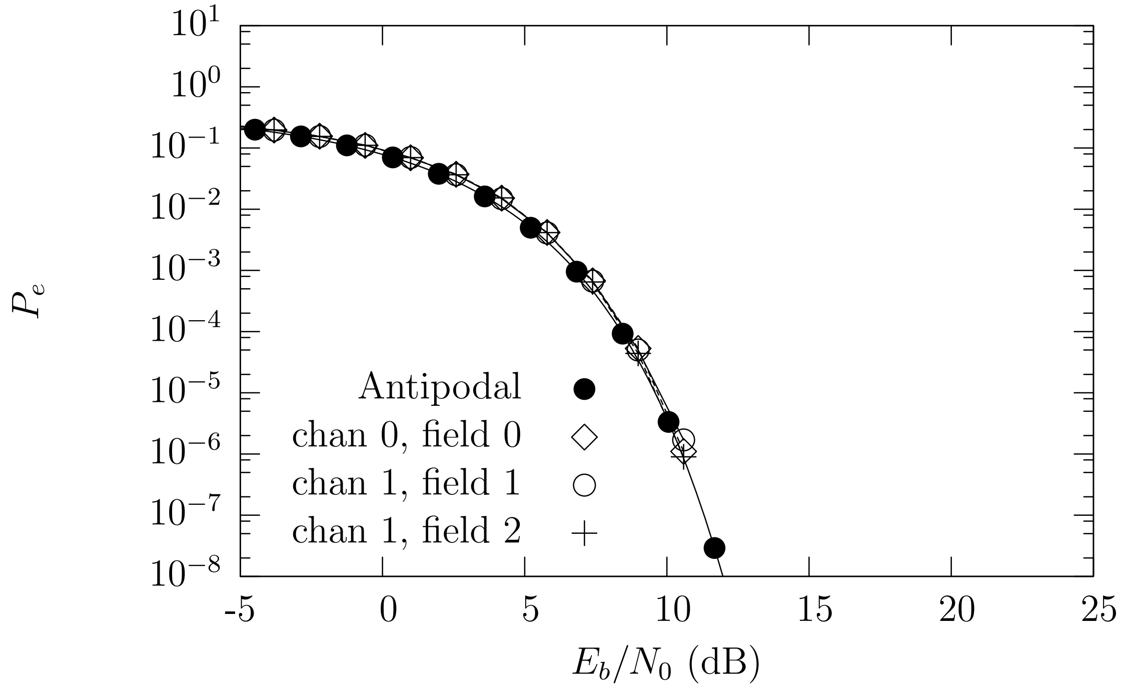


Figure 7.13: Case D: Joint Field-Based MAP Performance Curves for Various Field Numbers Correlation Coefficient $\rho = 1$, asynchronous case $\Delta\tau = 0.5T_s$

3. Case F:

In this next example the results for a detector with fixed width fields but variable distributions are presented. The signals, with structure illustrated in Table 7.15 and 7.16, have field widths identical to Case D presented in Section VII.H.2. As illustrated in Figure 7.14, this results in a much larger trellis. Figure 7.15 shows the same trellis enlarged and cropped for legibility.

Table 7.15: Case F: Signal 0

Field Number	Width	Values / Prior Pairs		
0	3	(001, 0.2)	(000, 0.5)	(101, 0.3)
1	3	(000, 0.125)	(001, 0.125)	(010, 0.125)
		(100, 0.125)	(101, 0.125)	(110, 0.125)
2	3	(000, 0.01)	(001, 0.01)	(010, 0.01)
		(100, 0.1)	(101, 0.6)	(110, 0.1)
3	3	(000, 0.5)	(111, 0.5)	

Table 7.16: Case F: Signal 1

Field Number	Width	Values / Prior Pairs		
0	3	(001, 0.5)	(000, 0.5)	
1	3	(001, 0.2)	(000, 0.2)	(011, 0.2)
2	3	(000, 0.125)	(001, 0.125)	(010, 0.125)
		(100, 0.125)	(101, 0.125)	(110, 0.125)
3	3	(001, 0.5)	(000, 0.5)	

a. ***Signal Offset*** $\Delta\tau = 0T_s$

Figure 7.16 presents the FER performance for Signal 0 Field 3 of the Joint Field-Based MAP Detector for synchronous signals and various cross-correlation coefficients. Note from Tables 7.15 and 7.16 that the two fields, Signal 0 Field 3 and Signal 1 Field 3, are further in Euclidean distance than is the case for the signals presented earlier. This increase in distance reduces the probability of field error.

b. ***Signal Offset*** $\Delta\tau = 0.5T_s$

Figure 7.17 presents the FER performance for Signal 0 Field 3 of the Joint Field-Based MAP Detector for asynchronous signals with a separation of $\Delta\tau = 0.5T_s$, and various cross-correlation coefficients. The performance here is slightly better than that

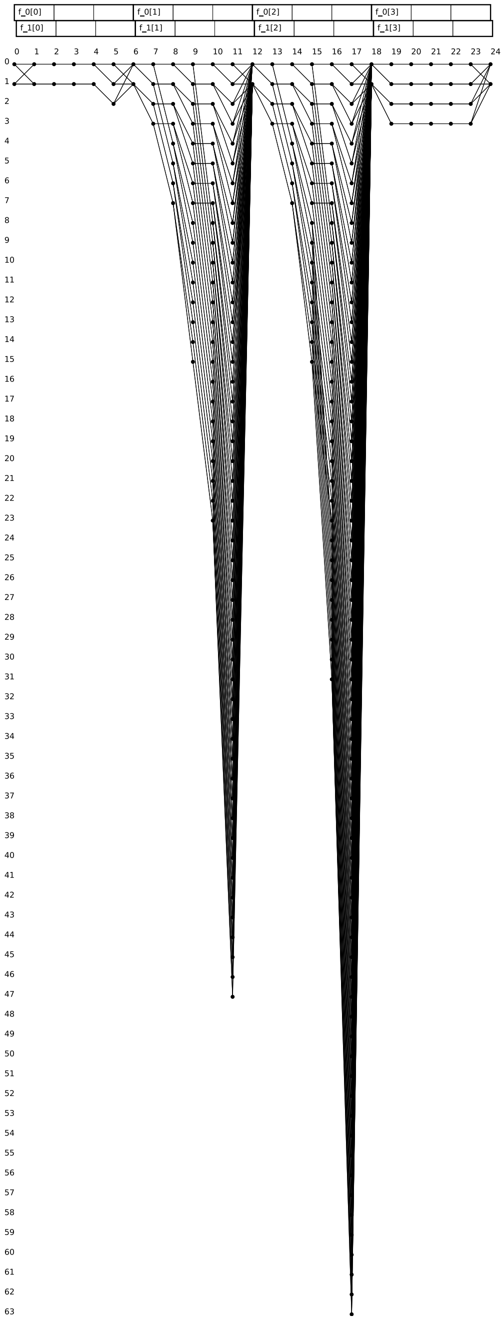


Figure 7.14: Case F: Full Trellis

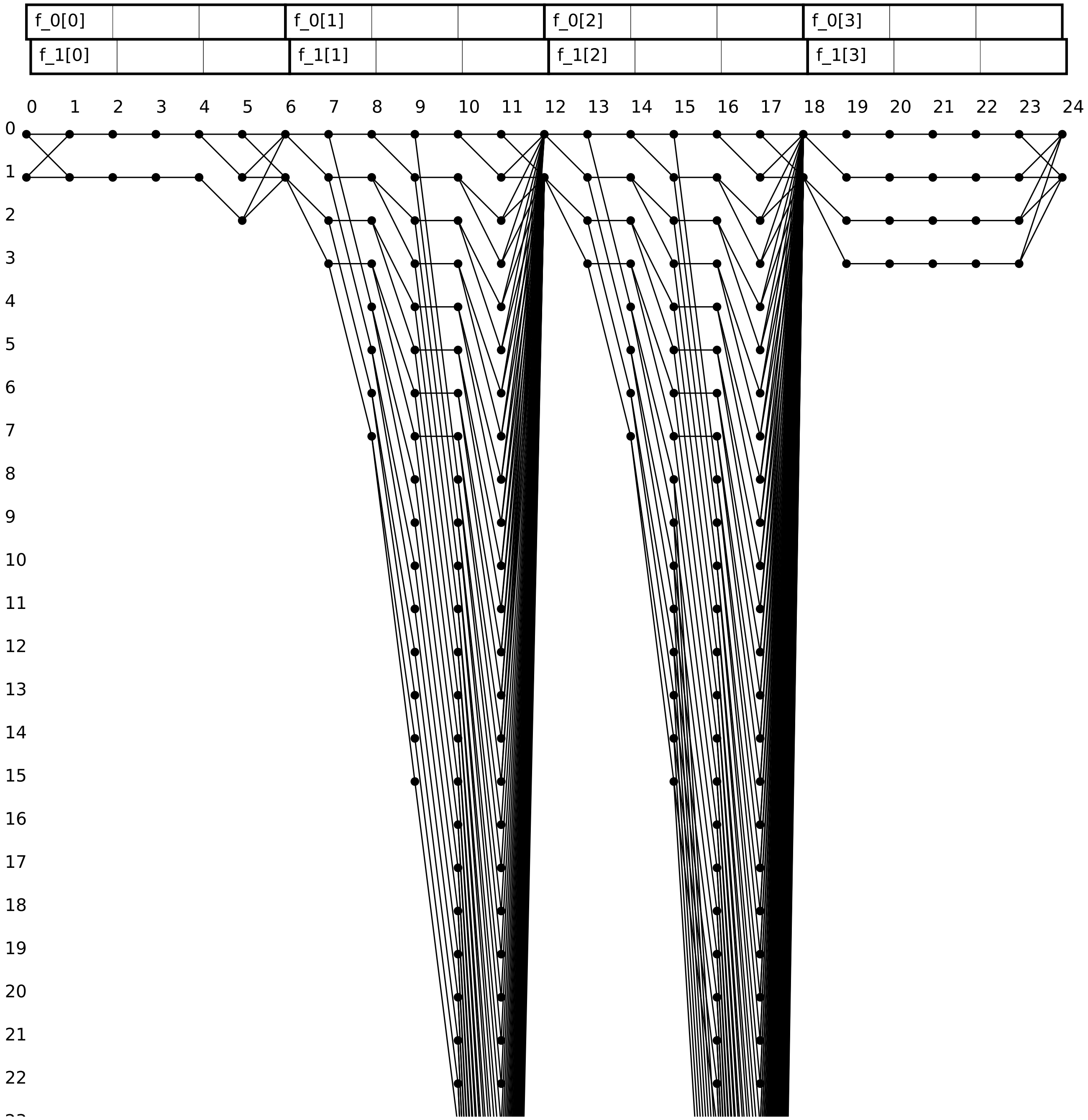


Figure 7.15: Case F: Trellis

Table 7.17: Case F: Number Of Elements Per Stage

Stage Number	Elements					
1	$a_0[0]$					
2	$a_0[0]$	$a_1[0]$				
3	$a_0[0]$	$a_1[0]$	$a_0[1]$			
4	$a_0[0]$	$a_1[0]$	$a_0[1]$	$a_1[1]$		
5	$a_0[0]$	$a_1[0]$	$a_0[1]$	$a_1[1]$	$a_0[2]$	
6	$a_1[2]$					
7	$a_1[2]$	$a_0[3]$				
8	$a_1[2]$	$a_0[3]$	$a_1[3]$			
9	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$		
10	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	$a_1[4]$	
11	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	$a_1[4]$	$a_0[5]$
12	$a_1[5]$					
13	$a_1[5]$	$a_0[6]$				
14	$a_1[5]$	$a_0[6]$	$a_1[6]$			
15	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$		
16	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$	$a_1[7]$	
17	$a_1[5]$	$a_0[6]$	$a_1[6]$	$a_0[7]$	$a_1[7]$	$a_0[8]$
18	$a_1[8]$					
19	$a_1[8]$	$a_0[9]$				
20	$a_1[8]$	$a_0[9]$	$a_1[9]$			
21	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$		
22	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$	$a_1[10]$	
23	$a_1[8]$	$a_0[9]$	$a_1[9]$	$a_0[10]$	$a_1[10]$	$a_0[11]$
24	$a_1[11]$					

of the synchronous case presented above. For this case the FER performance for each field will depend on the *a priori* distributions as outlined in Table 7.15 and 7.16. This variation is illustrated in Figure 7.18 for the synchronous case and in Figure 7.19 for the asynchronous case.

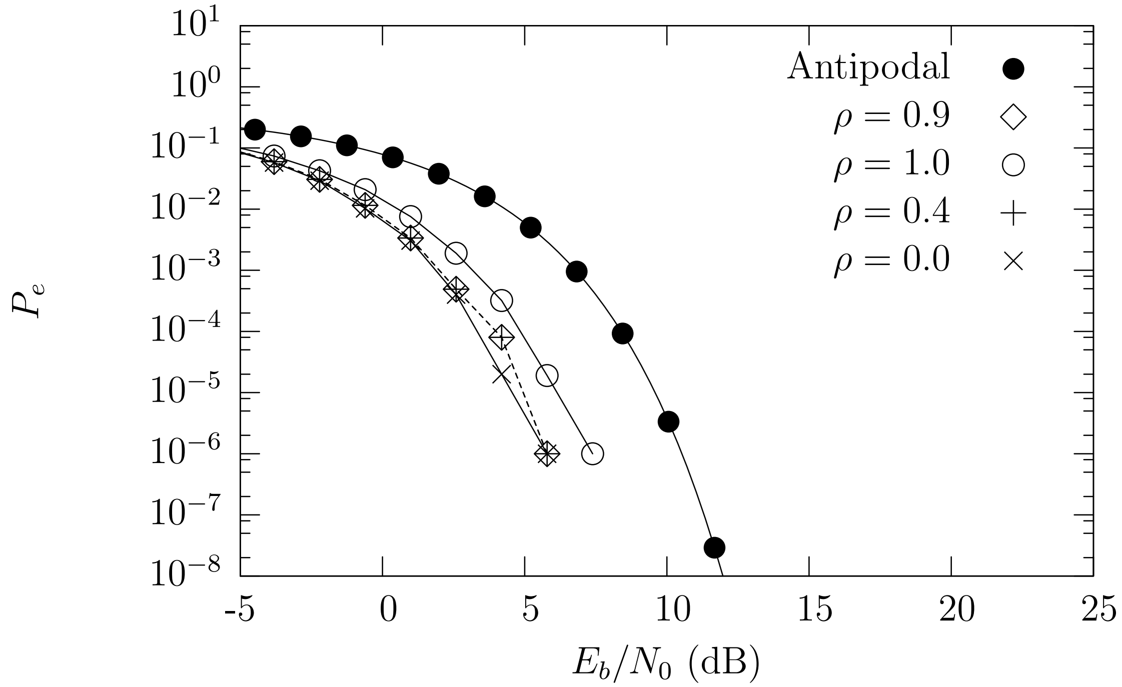


Figure 7.16: Case F: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, synchronous case $\Delta\tau = 0T_s$, fixed width fields

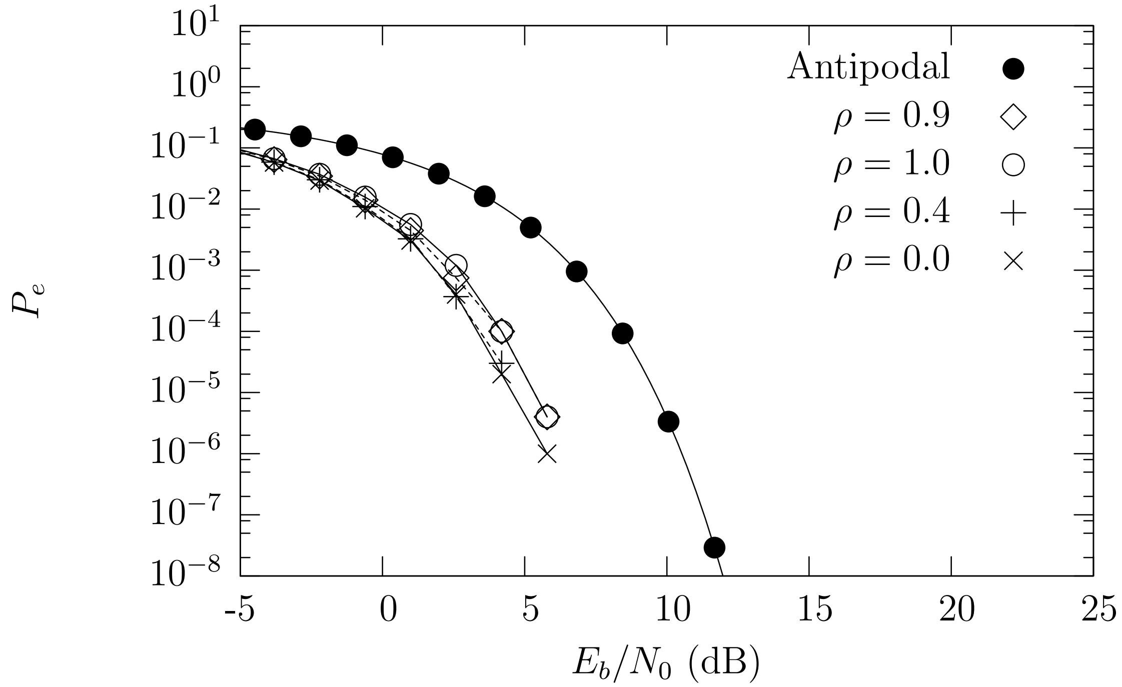


Figure 7.17: Case F: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, asynchronous case $\Delta\tau = 0.5T_s$

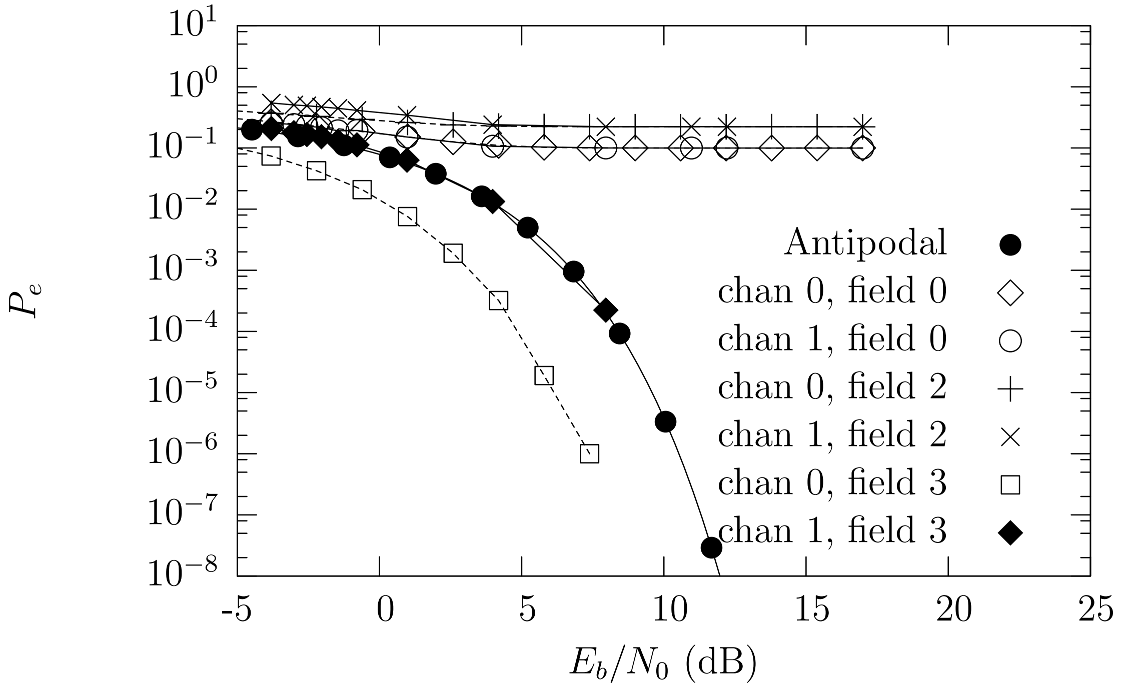


Figure 7.18: Case F: Joint Field-Based MAP Performance Curves for Various Field Numbers Correlation Coefficient $\rho = 1$, synchronous case $\Delta\tau = 0T_s$

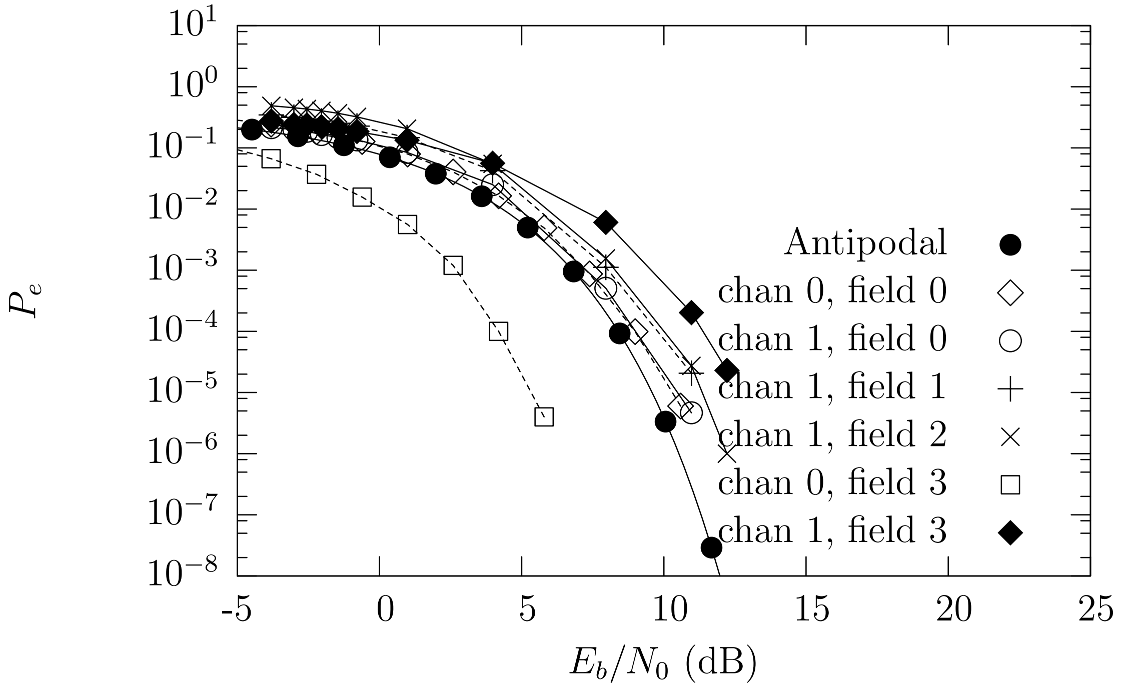


Figure 7.19: Case F: Joint Field-Based MAP Performance Curves for Various Field Numbers Correlation Coefficient $\rho = 1$, asynchronous case $\Delta\tau = 0.5T_s$

4. Case B:

In this final example the performance of the Joint Field-Based MAP Detector for signals with variable field widths is presented.

The structure of the two signals are shown in Table 7.18 and Table 7.19. Each signal consists of fields of various widths which take on various distributions. Notice that the structure of the trellis in Figure 7.20 changes for the different field widths and prior distributions.

Table 7.18: Case B: Signal 0

Field Number	Width	Values / Prior Pairs		
0	1	(0, 0.5)	(1, 0.5)	
1	1	(0, 0.5)	(1, 0.5)	
2	1	(1, 1)		
3	3	(000, 0.2)	(101, 0.6)	(111, 0.2)
4	1	(0, 0.5)	(1, 0.5)	
5	1	(0, 0.5)	(1, 0.5)	
6	1	(0, 0.5)	(1, 0.5)	

Table 7.19: Case B: Signal 1

Field Number	Width	Values / Prior Pairs	
0	1	(1, 0.5)	(0, 0.5)
1	1	(1, 0.5)	(0, 0.5)
2	2	(01, 0.5)	(00, 0.5)
3	1	(0, 0.5)	(1, 0.5)
4	2	(00, 0.5)	(11, 0.5)
5	1	(0, 0.5)	(1, 0.5)
6	1	(0, 0.5)	(1, 0.5)

a. *Signal Offset* $\Delta\tau = 0T_s$

In this first example, FER curves are plotted for one of the fields for the synchronous case and various values for the cross-correlation coefficient ρ (7.3).

Figure 7.21 illustrates the FER performance for Chan 0 Field 3. The performance here is slightly better than some of the previous cases.

Table 7.20: Case B: Number Of Elements Per Stage

Stage Number	Elements				
1	$a_0[0]$				
2	$a_1[0]$				
3	$a_0[1]$				
4	$a_1[1]$				
5	$a_0[2]$				
6	$a_0[2]$	$a_1[2]$			
7	$a_0[2]$	$a_1[2]$	$a_0[3]$		
8	$a_0[2]$	$a_1[2]$	$a_0[3]$	$a_1[3]$	
9	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	
10	$a_1[2]$	$a_0[3]$	$a_1[3]$	$a_0[4]$	$a_1[4]$
11	$a_0[5]$				
12	$a_0[5]$	$a_1[5]$			
13	$a_0[5]$	$a_1[5]$	$a_0[6]$		
14	$a_1[6]$				
15	$a_0[7]$				
16	$a_1[7]$				
17	$a_0[8]$				
18	$a_1[8]$				

b. *Signal Offset* $\Delta\tau = 0.5T_s$

As in Case A, with an offset between the two signals we can achieve reasonable FER performance even in the case of highly correlated signals; Figure 7.22 illustrates this. Also, notice in Figure 7.23 there is not a large variation in performance based on field number.

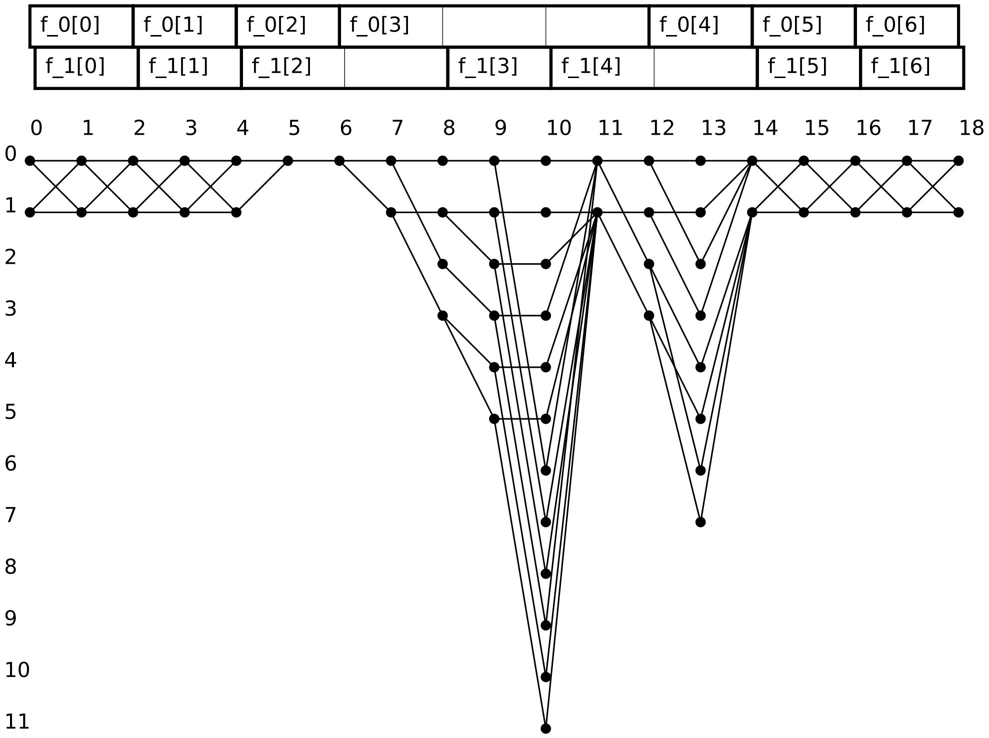


Figure 7.20: Case B: Trellis

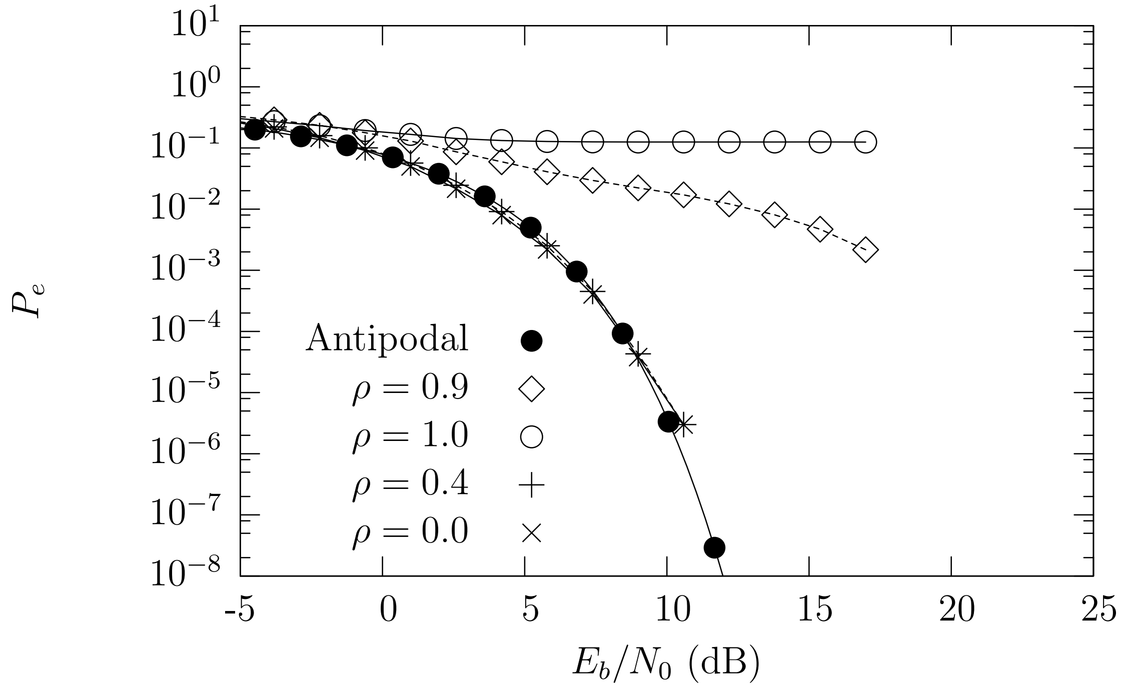


Figure 7.21: Case B: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, synchronous case $\Delta\tau = 0T_s$

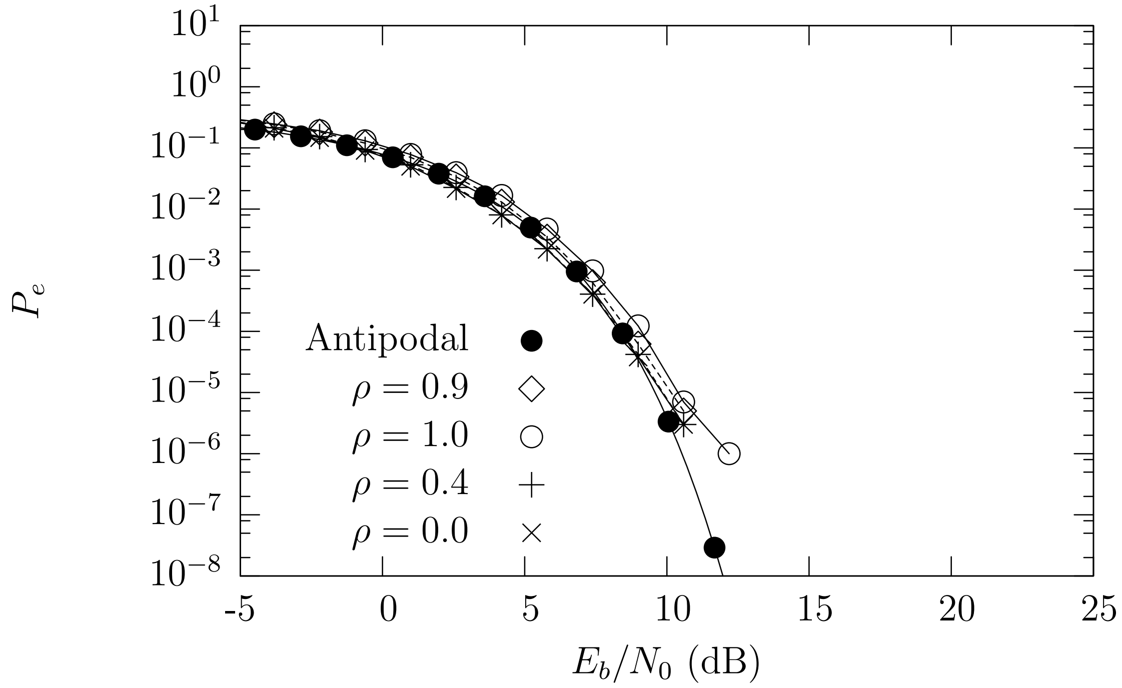


Figure 7.22: Joint Field-Based MAP Performance Curves for Various Values of Correlation Coefficient ρ, asynchronous case $\Delta\tau = 0.5$

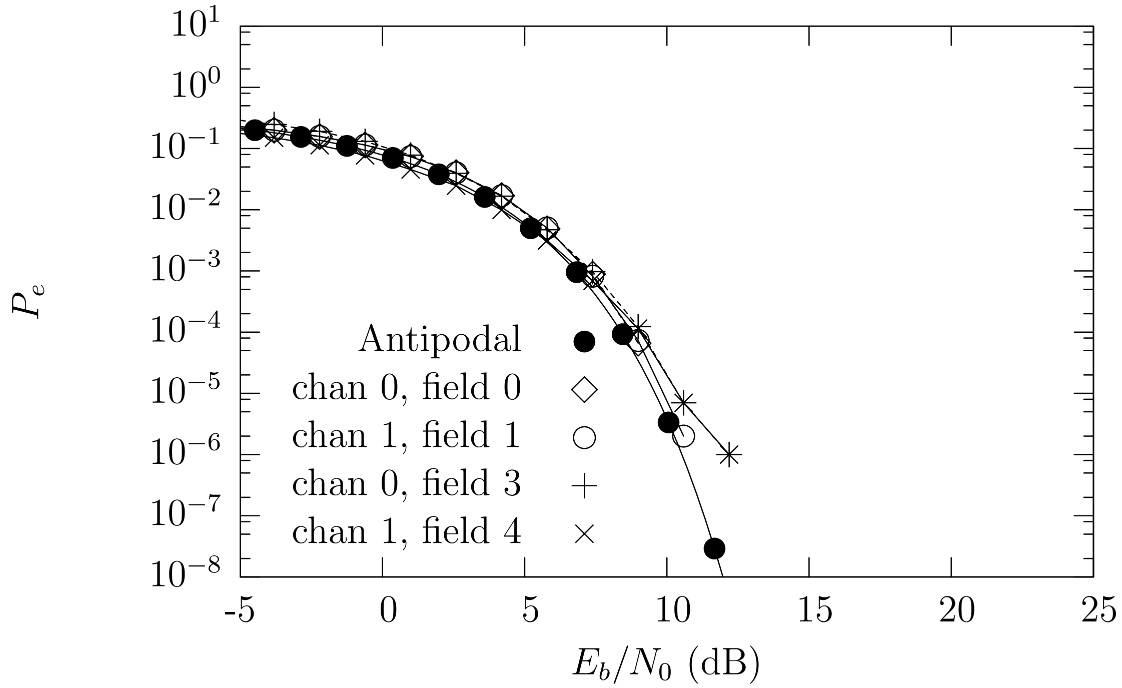


Figure 7.23: Joint Field-Based MAP Performance Curves Variation Based on Distance from Start of Packet Correlation Coefficient $\rho = 1$, asynchronous case $\Delta\tau = 0.5T_s$

I. CHAPTER SUMMARY

The Joint Field-Based MAP Detector is introduced in this chapter. This detector optimally incorporates *a priori* knowledge of variable width field values to aid in joint detection. The Joint Field-Based MAP detector can also be applied to situations where knowledge of one of the cochannel signals is very limited. If there is no knowledge about the field structure of one of the signals the knowledge of the other signals can still be exploited to aid in recovery. The signal model is developed before and after the whitened matched filter. We then developed the form of the trellis required to use the BCJR algorithm to generate the MAP estimates. Finally results are presented for nominal scenarios investigating how the performance varies as the structure of the signals change. These results demonstrate the potential performance improvement that is possible by incorporating *a priori* information into a joint detection algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSIONS AND FUTURE WORK

The goal of this work is to investigate methods for efficient recovery of the information content from low-entropy narrowband cochannel signals. There is significant prior work in joint detection of narrowband signals [2], but little work has focused on low-entropy signals. In this dissertation, optimal (minimum probability of field error) methods are developed to achieve this goal. A summary of the contributions from this work and suggestions for continued research are presented here.

A. SUMMARY

In Chapter II, the fundamentals of multi-user detection required for describing joint detection are described. A detailed background of the signaling waveform used in AIS (GMSK) is provided. A whitened discrete model of the signal is developed using the Laurent representation of GMSK. The MAP algorithm is then introduced as an efficient method for calculating *a posteriori* symbol probabilities. A literature review of joint detection methods is presented, focusing on work by Verdú [2].

In Chapter III, a complete overview of the AIS is presented. The signaling waveform and the packet structure of the AIS are presented. A link budget is presented demonstrating that for nominal conditions, a receiver collecting from low earth orbit (approximately 800 km) will not be noise-limited. An example of non-coherent AIS demodulation is presented showing each step required to recover the transmitted data. The low-entropy property of the AIS message is demonstrated with an example. A 168-bit AIS packet is found to typically contain approximately 20 bits of new information. Finally the SOTDMA algorithm is investigated, and the predictability of subsequent AIS transmissions is presented.

Chapter IV introduces methods of prediction focusing on AIS field values. Three different candidate predictors for the AIS fields are presented, and tuning parameters for the predictors are heuristically developed. Further, a novel predictor for the AIS communications state field is presented, and a demonstration of the performance of a Kalman filter on AIS latitude and longitude fields.

In Chapter V, MLSD is investigated for a bit-stuffed data source. The bit-stuffing process is modeled in a form compatible with a trellis representation. With this representation, the Viterbi algorithm may be used to find efficiently the most likely sequence.

The bit-stuffing process is found to be a catastrophic encoding process (a single bit error leads to a large number of decoder errors). Approximate BER bounds for MLSD of a bit-stuffed data source are developed. The MLSD is shown to have superior BER performance compared to symbol-by-symbol decisions (a gain of 0.2 – 1.0 dB for the unstuffed data).

The Field-Based MAP detector is developed in Chapter VI. The performance of this detector is demonstrated using representative data. The detector is shown to outperform the symbol-by-symbol ML detector for some example *a priori* distributions. The detector is extended to multiple synchronous signals, and the performance of the resulting detector is presented. The performance gains from Field-Based MAP detection are highly dependent on the level of *a priori* information available and, in the joint case, on the correlation coefficient. The examples in Chapter VI demonstrate a performance improvement of 3 to 18 dB of gain when using joint Field-Based MAP compared to joint symbol-by-symbol detection.

Finally, Chapter VII develops the Joint Field-based MAP detector for asynchronous signals. The algorithms provided in this chapter illustrate the construction of a trellis compatible with the BCJR algorithm ultimately allowing for the calculation of *a posteriori* probabilities for transmitted field values. An efficient implementation of the Joint Field-Based MAP detector is developed and field error rate performance is investigated. The performance gains here are identical to that of the previous chapter for the synchronous case ($\Delta\tau = 0$). For asynchronous signals the results from Chapter VI are not applicable, and the detector from Chapter VII must be used. For the asynchronous case there are additional FER performance gains of up to 10 dB, depending heavily on the cross-correlation between the two signals.

B. CONCLUSIONS

Figure 8.1 gives a “big picture” view of the objective in this work as applied to remote reception of the AIS. The algorithm begins with some knowledge about existing ships — this could be very specific information such as a previous position report or very general, such as “Ship A left Port B at 1300 yesterday.” At the receiver, two or more signals are received overlapping in frequency and time. The signal parameters for each of the overlapping signals are estimated. This reception is passed through multiple matched filters, and each filter is matched to a specific pulse with estimated parameters. The

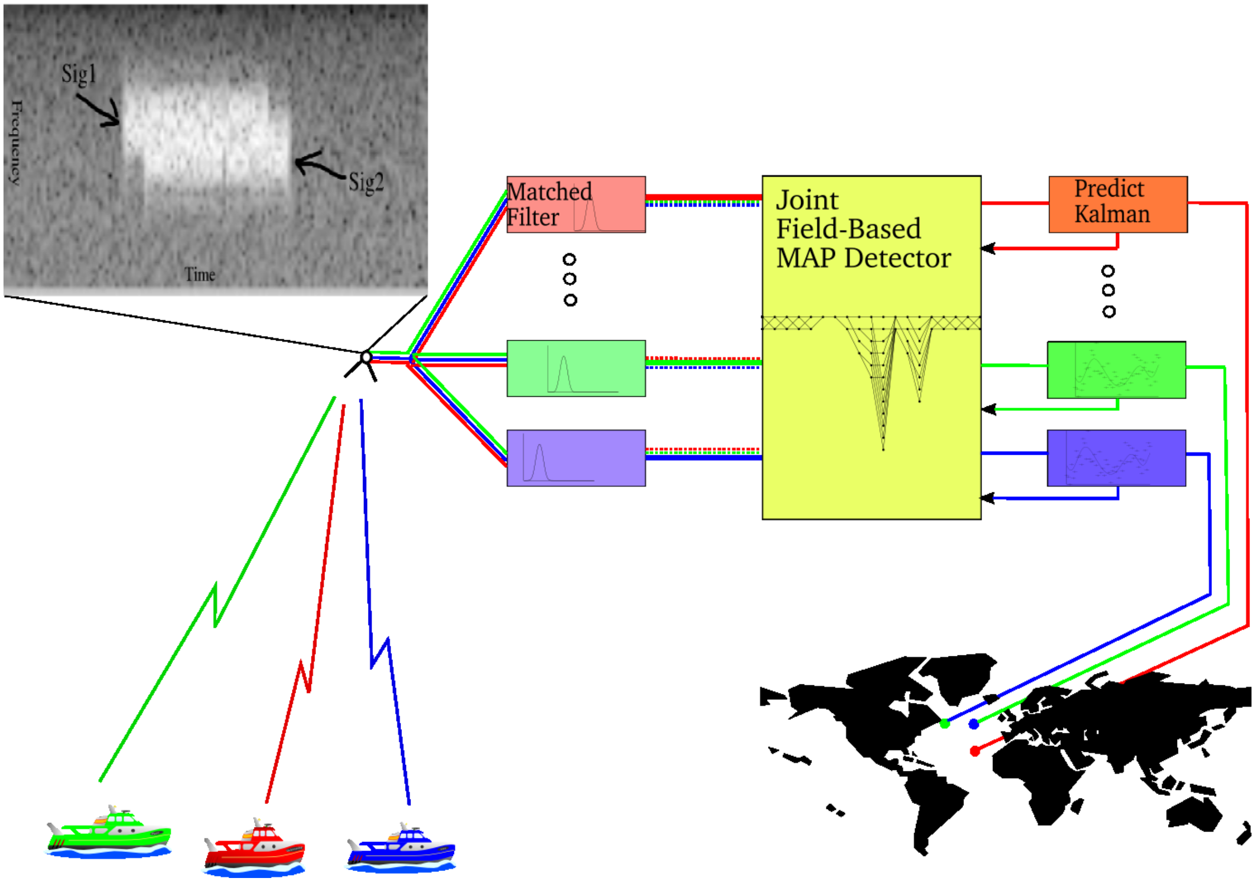


Figure 8.1: "Big Picture"

resulting decision statistics are then passed to the Joint Field-Based MAP detector, along with *a priori* information derived from previous receptions or other sources. The output of the decisions from the joint detector are then passed to prediction algorithms for future receptions. This dissertation develops the important pieces for achieving this goal: the joint field-based MAP detector, the prediction method for the AIS fields, and the MLSD for bit-stuffing.

The method depicted in Figure 8.1 is relevant even without *a priori* information for each of the simultaneous users. For example, suppose from previous transmissions it is known that Ship A will transmit on time slot 7. Based upon the reception at time slot 7, it is determined that there is an additional unknown ship, Ship X, occupying the same slot. With no knowledge of what Ship X is transmitting the field structure is simply set to equally likely symbols. The algorithm described in Chapter VII is now applied as before.

Many ideas from this work may be incorporated into receivers immediately. The work on MLSD in Chapter V is applicable whenever the source of the data has undergone bit-stuffing. If a receiver is already using the Viterbi algorithm for detection, the ideas from Chapter V can be incorporated by modifying the existing trellis. If the receiver is currently making symbol-by-symbol decisions, the Viterbi algorithm outlined in Chapter V may be implemented directly. The work in Chapter V demonstrates that a gain in BER performance of approximately 0.2 – 1.0 dB can be expected by using these techniques.

The work in Chapter VI can be readily extended to any problem where *a priori* information exists for sets of symbols. This is applicable to both single- and multiple-channel receivers. Both the gain in performance and the additional computational complexity will be very dependent on the particular situation. For example, if it is known that a set of symbols can take on very few values, then the computational complexity of implementing the field-based MAP detector is relatively low. If the set of values for a field of symbols are all separated by something greater than the minimum distance, then the performance improvement will likely be significant. If the set of possible values is very large, and the distance between pairs of values is approximately the minimum distance, then the small performance improvement may not justify the large computational burden of the field-based MAP detector.

Chapter VII presents a culmination of the ideas from previous chapters. A method for calculating *a posteriori* probabilities for a transmitted field is developed. In order to implement this method of signal recovery the trellis must be set up according to the known structure and *a priori* information of the signals. The structure of the trellis will vary as the information available for the signals changes. For the general case of asynchronous reception the method illustrated in Chapter VII generates *a posteriori* field probabilities leading to the optimal (minimum probability of field error) detector. Gains of up to 10 dB are seen compared to the synchronous joint detector presented in Chapter VI.

The theoretical work developed in Chapters VI and VII can be put to practical use as a bound for system performance. The detectors developed are optimal given the stated constraints. The FER curves from these chapters can be used as an upper bound on performance.

C. FUTURE WORK

In performing this research, many areas deserving additional investigation have been identified. Most of the future work involves solving practical problems. The recommendations can be roughly broken into three categories: FER performance improvement, parameter estimation, and computation reduction.

The work has been based on the first term of the Laurent decomposition. The trellis could be extended to use all the pulses, increasing performance. For simplicity, it is assumed there is no ISI between pulses; this is not true; GMSK introduces ISI. The performance demonstrated in this dissertation is based on a simplified channel. It is a straightforward extension to account for the ISI by adding states to the trellis. Furthermore, the MLSD presented for a bit-stuffed data source in Chapter V could be combined with the joint field-based maximum *a posteriori* detector (JFBMAPD), yielding even better FER performance.

For signals that are received with low SNR, summing the soft outputs of a MAP detector may be used to increase the likelihood of successful demodulation. This is perhaps more relevant to single-channel low-entropy signals.

In the area of parameter estimation, additional areas for further investigation include joint carrier-phase and frequency tracking methods. This work assumes a source of channel state information (phase and frequency estimates). Determining this is non-trivial. In theory, these parameters could be treated as additional unknown parameters in the trellis.

In the area of computation reduction, there are a few areas for further investigation. The optimum detector could be simplified by only searching back one symbol. This is similar to compressing the priors to one symbol. Looking back only one symbol is not optimal, but it would be interesting to compare the performance to Joint Field-based BCJR. There are also many variations on the BCJR algorithm that involve simplifying the computation [109].

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. LAURENT REPRESENTATION

The ability to represent a signal as a linear combination of amplitude modulated pulses greatly simplifies receiver analysis and design. It allows one to build on the theory available for pulse amplitude modulated signals.

Continuous-phase modulation (CPM) signals are in general nonlinear. CPM signals, in general, can not be represented as superposition of amplitude modulated pulses. Laurent [21] shows that for a finite length pulse response $g(t)$, any CPM signal can be written as a superposition of $M = 2^{L-1}$ amplitude modulated pulses, where L is the duration of the finite length pulse, $g(t)$, divided by the symbol period. This is an extension of the well known representation of MSK as offset quadrature phase shift keying (OQPSK). We begin with the derivation of the Laurent representation and then follow with two examples.

A. THE LAURENT REPRESENTATION

The discussion below follows from examples in [21, 23]. We start with the complex envelope representation of a continuous phase modulated signal

$$z(t) = A \exp\left(j \left[\theta_0 + 2\pi h \sum_n \alpha_n q(t - nT_s) \right] \right) \tag{A.1}$$

where h is the modulation index, θ_0 is an arbitrary constant phase term (assumed 0 hereafter), and A is a constant amplitude term (assumed 1 hereafter). The symbol $\alpha_n \in \{-1, 1\}$ contains the information for the n^{th} bit. The function $q(t)$ is the integrated finite duration pulse response

$$q(t) = \begin{cases} 0 & t < 0 \\ \int_0^t g(\tau)\, \mathrm{d}\tau & 0 \le t \le LT_s \\ 1/2 & t > LT_s \end{cases} \tag{A.2}$$

for integer L and symbol duration T_s. For GMSK, $g(t)$ is a Gaussian pulse shape. For the case of AIS operating in the "high" setting with a time-bandwidth product of $BT_s = 0.4$, L can safely be set to 4 without significant loss of energy.

Now we rewrite Equation (A.1) by extracting the phase contribution from all but the last L symbols. We then write the contribution of the last L symbols as the product of exponentials.

$$z(t) = \exp\left(j\pi h \sum_{n=-\infty}^{N-L} \alpha_n\right) \times \prod_{i=0}^{L-1} \exp\left(j2\pi h\left[\alpha_{N-i} q(t-(N-i)T_s)\right]\right) \tag{A.3}$$

The next step involves rewriting the complex exponential associated with the n^{th} symbol as the sum of two terms with only the second term depending on the n^{th} symbol α_n

$$\begin{aligned}\exp\left(j2\pi h\left[\alpha_n q(t-nT_s)\right]\right) =& \frac{\sin(\pi h - 2\pi h q(t-nT_s))}{\sin(\pi h)} \\ &+ \exp(j\pi h\alpha_n)\frac{\sin(2\pi h q(t-nT_s))}{\sin(\pi h)}\end{aligned} \tag{A.4}$$

This can be done by recognizing we want the form

$$\sin(\pi h - \beta(t-nT)) + \gamma\sin(\beta(t-nT)) \tag{A.5}$$

and using trigonometric identities to solve for the coefficients γ and β. We then define a symmetric generalized phase pulse:

$$\Psi(t) = \begin{cases} 2\pi h q(t) & t < LT_s \\ 2\pi h(1/2 - q(t-LT_s)) & LT_s \le t \end{cases} \tag{A.6}$$

For notational convenience we define the function

$$S_n(t) = \frac{\sin[\Psi(t+nT_s)]}{\sin(\pi h)} = S_0(t+nT_s) \tag{A.7}$$

We can now substitute Equation (A.6) and (A.7) in Equation (A.4) to obtain

$$\exp\left(j\left[\alpha_n\Psi(t-nT_s)\right]\right) = S_{L-n}(t) + \exp(j\pi h\alpha_n)S_{-n}(t) \tag{A.8}$$

We can now use Equation (A.8) in Equation (A.3) to get

$$z(t) = \exp\left(j\pi h \sum_{n=-\infty}^{N-L} \alpha_n\right) \prod_{i=0}^{L-1}\left[S_{i+L-N}(t) + \exp(j\pi h\alpha_{N-i})S_{i-N}(t)\right] \tag{A.9}$$

The L product terms on the right side of Equation (A.9) can be expanded to a sum of 2^L product terms, (actually 2^{L-1} from symmetry). Following the notation in [22] we introduce the index $K \in \{0, \ldots, 2^{L-1} - 1\}$. We use the notation $K(i) \in \{0, 1\}$ to represent the i^{th} bit in the binary representation of K

$$K = \sum_{i=0}^{L-1} K(i)2^i \tag{A.10}$$

For example if $L = 3$ and $K = 3$, then $K(0) = 1$, $K(1) = 1$, and $K(2) = 0$.

We now group these terms defining what is commonly referred to as the Laurent pulses.

$$C_K(t) = S_0(t) \times \prod_{i=1}^{L-1} S_{i+K(i)L}(t) \tag{A.11}$$

Observe the Laurent pulses $C_K(t)$ are formed from a product of shifted $S_0(t)$ pulses. We then use the $K(i)$ notation to define the complex phase coefficients

$$A_{0,N} = \sum_{n=-\infty}^{N} \alpha_n \tag{A.12}$$

and

$$A_{K,N} = A_{0,N} - \sum_{i=1}^{L-1} K(i)\alpha_{N-i} \tag{A.13}$$

and express the original CPM signal as a sum of amplitude modulated pulses

$$z(t) = \sum_{N=-\infty}^{\infty} \sum_{K=0}^{2^{L-1}-1} \exp(j\pi h A_{K,N}) C_K(t - nT) \tag{A.14}$$

B. EXAMPLES

In this section we provide an example of the Laurent representation for MSK and an example of the Laurent representation for GMSK as used in AIS as defined in [1] using a reduced number of terms.

1. Laurent Representation for MSK

The Laurent representation for MSK leads to the well known interpretation of MSK as OQPSK with half sine waves as pulses. For MSK, $h = 1/2$, $L = 1$, and the pulse $g(t)$ is given by

$$g(t) = \begin{cases} \frac{1}{2T_s} & 0 \leq t \leq T_s \\ 0 & \text{otherwise} \end{cases} \tag{A.15}$$

The integrated pulse term $q(t)$ is then

$$q(t) = \begin{cases} 0 & t \leq 0 \\ \frac{t}{2T_s} & 0 \leq t \leq T_s \\ 1/2 & \text{otherwise} \end{cases} \tag{A.16}$$

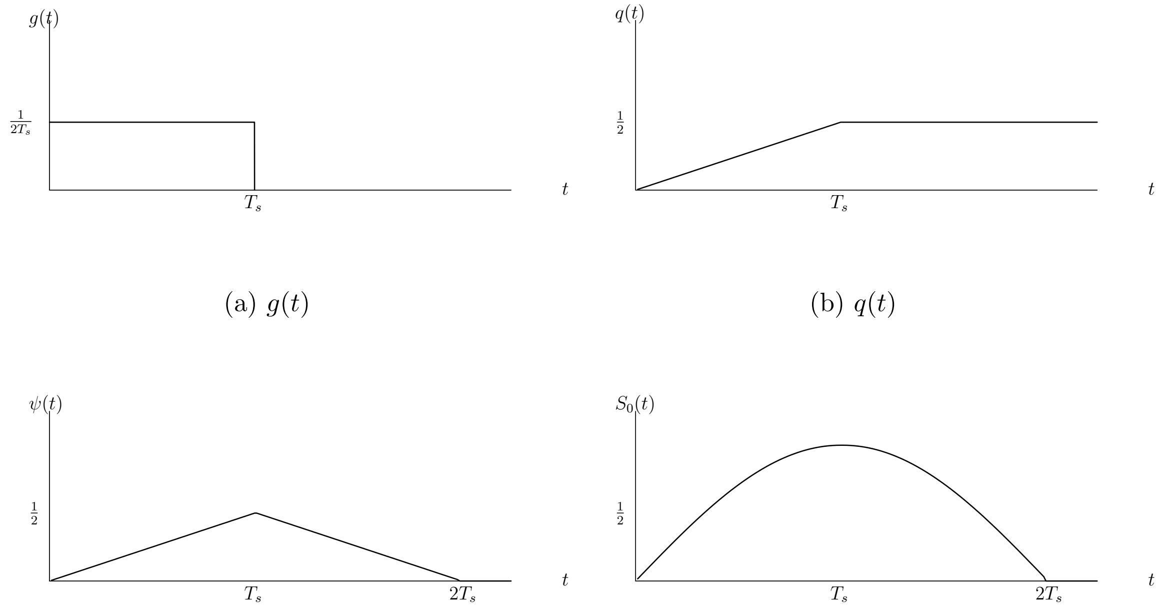
The symmetric generalized phase pulse $\Psi(t)$ is given by

$$\Psi(t) = \begin{cases} \frac{\pi t}{2T_s} & 0 \leq t < T_s \\ \frac{-\pi(t-2T_s)}{2T_s} & T_s \leq t \leq 2T_s \\ 0 & \text{otherwise} \end{cases} \tag{A.17}$$

We can use Equation A.7 to find S_0 as

$$S_0(t) = \begin{cases} \sin\left[\frac{\pi t}{2T_s}\right] & 0 \leq t \leq 2T_s \\ 0 & \text{otherwise} \end{cases} \tag{A.18}$$

Each of the above signals is shown in Figure A.1



(a) $g(t)$

(b) $q(t)$

(c) $\Psi(t)$

(d) $S_0(t)$

Figure A.1: Pulse shapes for MSK

For $L = 1$ the only Laurent pulse is the C_0 term $C_0(t) = S_0(t)$ and the only complex phase coefficient is $A_{0,N}$. So for the case of MSK

$$z(t) = \sum_{N=-\infty}^{\infty} \exp(j\pi h A_{0,N}) C_0(t - nT) \tag{A.19}$$

which is identical to the OQPSK representation as a superposition of half sine wave pulses.

2. Laurent Representation for AIS

The Laurent representation for GMSK as used in AIS is significantly more involved than the representation for MSK. We will show that for GMSK with $BT = 0.4$, $h = 0.5$, and $L = 4$, although 16 terms are required for exact representation, using only the first one or two terms results in minimal error.

For GMSK, $g(t)$ is a Gaussian pulse.

$$g(t) = \text{rect}(t/T_s) * \frac{1}{\sqrt{2\pi\sigma^2 T_s^2}} \exp\left(\frac{-t^2}{2\sigma^2 T_s^2}\right) \tag{A.20}$$

where $*$ denotes convolution, $\sigma = \sqrt{\ln(2)}/2\pi BT_s$ [20], and $\text{rect}(t)$ is defined as

$$\text{rect}(t) = \begin{cases} 1 & |t| < 1/2 \\ 0 & |t| > 1/2 \end{cases} \tag{A.21}$$

The pulse response $g(t)$ can be written as a difference of Q functions[1],

$$g(t) = \frac{1}{2T_s}\left[Q\left(2\pi BT_s \frac{t - T_s/2}{T_s\sqrt{\ln(2)}}\right) - Q\left(2\pi BT_s \frac{t + T_s/2}{T_s\sqrt{\ln(2)}}\right)\right] \tag{A.23}$$

where BT_s is the time-bandwidth product of the pulse response $g(t)$ and B is the 3dB bandwidth of the Gaussian filter.

Typically $g(t)$ will be truncated to length L depending on the value of BT_s such that $g(t)$ is zero outside the interval $[0, LT_s]$ and symmetric around $LT_s/2$.

The integrated pulse term $q(t)$ is then

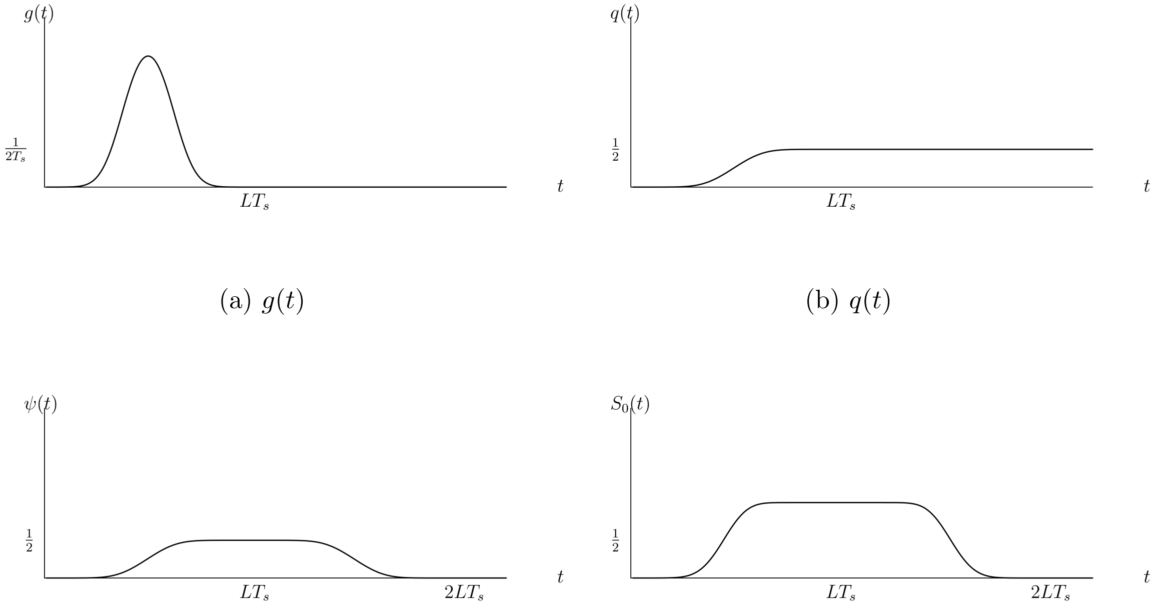
$$q(t) = \begin{cases} 0 & t < 0 \\ \int_0^t g(\tau)\,\mathrm{d}\tau & 0 \le t \le LT_s \\ 1/2 & t > LT_s \end{cases} \tag{A.24}$$

The function $\Psi(t)$ is given by

$$\Psi(t) = \begin{cases} 2\pi h q(t) & 0 \le t < LT_s \\ -2\pi h q(t - 2LT_s) & LT_s \le t \le 2LT_s \\ 0 & \text{otherwise} \end{cases} \tag{A.25}$$

[1] The Q-function is defined as

$$Q(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2}\,\mathrm{d}\tau \tag{A.22}$$



(a) $g(t)$

(b) $q(t)$

(c) $\Psi(t)$

(d) $S_0(t)$

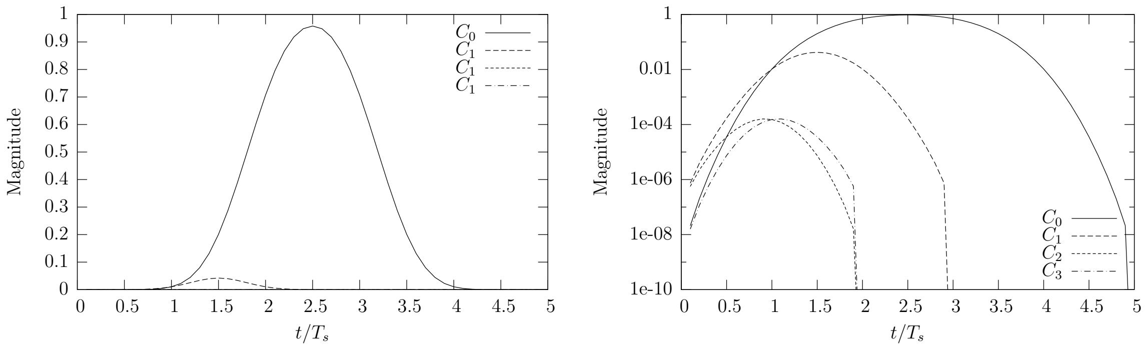
Figure A.2: Pulse shapes for GMSK

We can use Equation A.7 to find S_0

$$S_0(t) = \begin{cases} \sin(\Psi(t)) & 0 \leq t \leq 2LT_s \\ 0 & \text{otherwise} \end{cases} \tag{A.26}$$

Each of the above signals is shown in Figure A.2

For $L = 4$ there will be $M = 2^{L-1} = 8$ Laurent pulses $\{C_o(t), \ldots, C_7(t)\}$ which can be found by evaluating Equation A.11. For illustration, the product terms to form the 8



(a) Linear Scale

(b) Log Scale

Figure A.3: The first four pulses C_0,C_1,C_2, and C_3 for the Laurent representation with $L = 4$, $h = 0.5$, $BT_s = .4$

pulses are shown below.

$$\begin{aligned}
C_0(t) =& S_0(t) \times S_1(t) \times S_2(t) \times S_3(t) \\
C_1(t) =& S_0(t) \times S_5(t) \times S_2(t) \times S_3(t) \\
C_2(t) =& S_0(t) \times S_1(t) \times S_6(t) \times S_3(t) \\
C_3(t) =& S_0(t) \times S_5(t) \times S_6(t) \times S_3(t) \\
C_4(t) =& S_0(t) \times S_1(t) \times S_2(t) \times S_7(t) \\
C_5(t) =& S_0(t) \times S_5(t) \times S_2(t) \times S_7(t) \\
C_6(t) =& S_0(t) \times S_1(t) \times S_6(t) \times S_7(t) \\
C_7(t) =& S_0(t) \times S_5(t) \times S_6(t) \times S_7(t)
\end{aligned} \tag{A.27}$$

Figure A.3 illustrates the first four Laurent pulses. Notice that the first four Laurent pulses $\{C_0, C_1, C_2, C_3\}$ contain 99.905%, 0.095%, 0.000014%, 0.000014% of the energy respectively.

The complex phase coefficients are calculated with Equation A.12 and A.13. For illustration, the 8 complex phase coefficients are shown below.

$$
\begin{aligned}
A_{0,N} &= \sum_{n=-\infty}^{N} \alpha_n \\
A_{1,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-1} \\
A_{2,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-2} \\
A_{3,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-1} - \alpha_{N-2} \\
A_{4,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-3} \\
A_{5,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-1} - \alpha_{N-3} \\
A_{6,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-2} - \alpha_{N-3} \\
A_{7,N} &= \sum_{n=-\infty}^{N} \alpha_n - \alpha_{N-1} - \alpha_{N-2} - \alpha_{N-3}
\end{aligned}
\tag{A.28}
$$

The complex part of $z(t)$ formed using three methods: an integrated phase term, only the first Laurent pulse, and finally the first and second Laurent pulse. This is shown in Figure A.4. Notice that using only the first two pulses allows us to match that of the integrated phase term with a mean squared error (MSE) of only 2.3×10^{-6} %. Using only the first Laurent pulse results in a MSE of only 9.4×10^{-2} %.

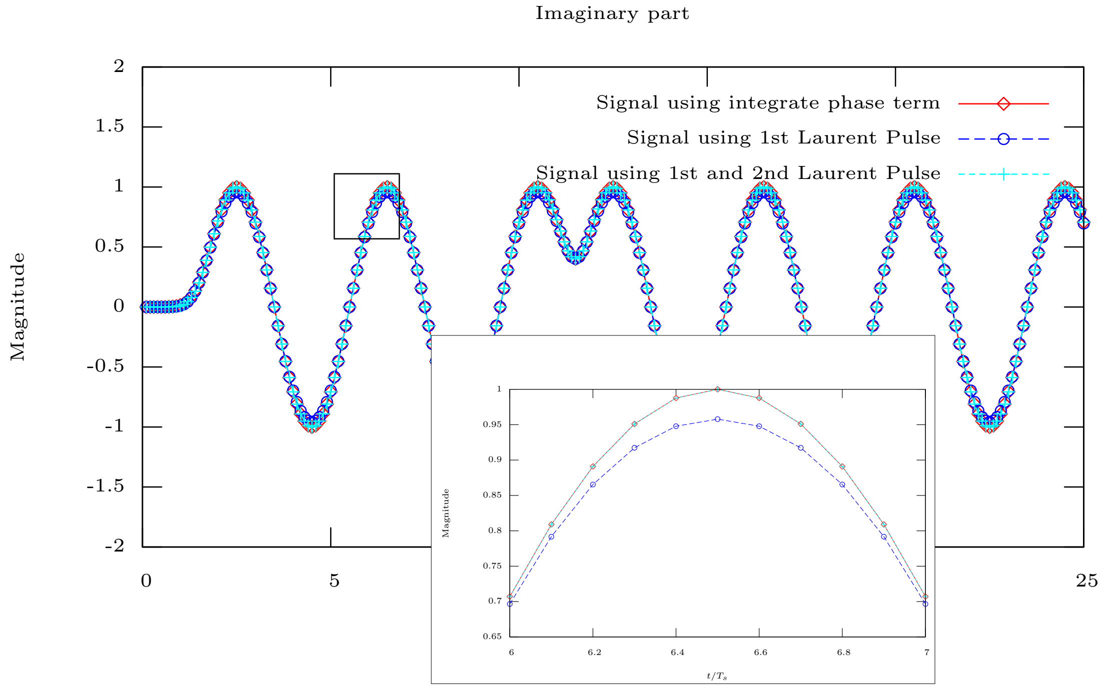


Figure A.4: GMSK signal generated with integrated phase term, first Laurent pulse and then the first and second Laurent pulse

APPENDIX B. BCJR ALGORITHM

The MAP algorithm is an efficient algorithm for estimating the *a posteriori* probabilities of the state transitions of a Markov source. The MAP algorithm, also called BCJR after the authors Bahl, Cocke, Jelinek and Raviv was originally proposed in [83]. Consider a sequence of values $a \in \mathcal{A}$ where $\mathcal{A}$ is a finite alphabet. The values a are then mapped one-to-one to symbols α and passed through a channel with finite impulse response. We can write the output of a whitened matched filter for this channel as

$$y[n] = \sum_{k=0}^{L-1} \alpha[n]h[k-n] + z[n] \tag{B.1}$$

where L is the length of the equivalent discrete-time channel h, and z is sampled bandlimited AWGN with variance σ^2. The MAP algorithm can be used to calculate the *a posteriori* probability $\Pr(a = a_x|\boldsymbol{y})$, where a_x is a specific element in $\mathcal{A}$.

Note: Throughout this dissertation, α has been used as above to refer to the transmitted symbol. For the remainder of this appendix α is used to represent an auxiliary probability function following the standard notation for BCJR.

The MAP algorithm is similar to the Viterbi algorithm in that it takes into account an entire received sequence in making decisions. The key difference is the MAP algorithm also incorporates *a priori* information of the transmitted symbols and the outputs of the MAP algorithm are *a posteriori* probabilities for each symbol. Where Viterbi selects the maximum likelihood sequence given a received sequence, MAP generates the *a posteriori* probabilities for each symbol given a received sequence.

The assumption for BCJR is the source can be written as a Markov process. Let $\lambda[n]$ be the state of the underlying Markov process at time n. The next state $\lambda[n+1]$ is determined by the value of the input and the value of the previous state $\lambda[n]$. If we know the starting state and all the inputs, we know all the state transitions. Similarly, if we know the state transitions, we know the inputs.

Given a source that generates independent symbols with a set of known *a priori* probabilities passed through a memoryless channel, the MAP algorithm provides no additional benefit from a symbol-by-symbol MAP detector. The MAP algorithm allows one to take advantage of the dependency among adjacent symbols. In the development of the BCJR algorithm below we draw on examples in [83], [109], and [108].

In our model, the "encoder" is a FIR filter. We encode the states such that the Markov property is maintained; i.e., knowledge of the state at time $n+1$ makes knowledge of the state at time n or previous times irrelevant.

We start by rewriting the received vector $\boldsymbol{y}$ as the union of prior, current, and future observations. This allows us to exploit the Markov property to simplify the calculation of the posterior probabilities.

$$\boldsymbol{y} = \boldsymbol{y}_{<n} \cup y_n \cup \boldsymbol{y}_{>n} \tag{B.2}$$

where $\boldsymbol{y}_{<n}$, y_n, and $\boldsymbol{y}_{>n}$ indicate the prior, current and future observations respectively.

We now define some auxiliary probability functions for notational convenience.

$$\alpha_n(p) = \Pr(\lambda[n] = p, \boldsymbol{y}_{<n}) \tag{B.3}$$

Equation (B.3) is the probability of being in state p at time n and all of the received values prior to n.

$$\beta_{n+1}(q) = \Pr(\boldsymbol{y}_{>n} | \lambda[n+1] = q) \tag{B.4}$$

Equation (B.4) is the probability of the received sequence of values received after n given the state at time $n+1$ is q.

$$\gamma_n(p,q) = \Pr(\lambda[n+1] = q, y_n | \lambda[n] = p) \tag{B.5}$$

Equation (B.5) is the probability of the observation at time n and the state being q at time $n+1$ given a previous state at time n of p.

Using Equation (B.2) and the identity $\Pr(A|B) = \Pr(A,B)/\Pr(B)$ we can write the posterior probability as

$$\Pr(\lambda[n] = p, \lambda[n+1] = q | \boldsymbol{y}) = \Pr(\lambda[n] = p, \lambda[n+1] = q, \boldsymbol{y}_{<n}, \boldsymbol{y}_n, \boldsymbol{y}_{>n}) / \Pr(\boldsymbol{y}) \tag{B.6}$$

We now factor (B.6) using the identity $\Pr(A,B)=\Pr(A)\Pr(B|A)$ to get

$$\begin{aligned}\Pr(\lambda[n]=p,\lambda[n+1]=q|\boldsymbol{y})=&\Pr(\lambda[n]=p,\lambda[n+1]=q,\boldsymbol{y}_{<n},y_n)\times\\&\Pr(\boldsymbol{y}_{>n}|\lambda[n]=p,\lambda[n+1]=q,\boldsymbol{y}_{<n},y_n)/\Pr(\boldsymbol{y})\end{aligned}\tag{B.7}$$

The first term in (B.7) can be factored using the same identity to get

$$\begin{aligned}\Pr(\lambda[n]=p,\lambda[n+1]=q|\boldsymbol{y})=&\Pr(\lambda[n]=p,\boldsymbol{y}_{<n})\times\\&\Pr(\lambda[n+1]=q,y_n|\lambda[n]=p,\boldsymbol{y}_{<n})\times\\&\Pr(\boldsymbol{y}_{>n}|\lambda[n]=p,\lambda[n+1]=q,\boldsymbol{y}_{<n},y_n)/\Pr(\boldsymbol{y})\end{aligned}\tag{B.8}$$

The Markov property says that knowledge of the state at time n renders knowledge of previous states and previous inputs irrelevant. We can use this property on the second and third factors on the right hand side of (B.8) to get

$$\begin{aligned}\Pr(\lambda[n]=p,\lambda[n+1]=q|\boldsymbol{y})=&\Pr(\lambda[n]=p,\boldsymbol{y}_{<n})\times\\&\Pr(\lambda[n+1]=q,y_n|\lambda[n]=p)\times\\&\Pr(\boldsymbol{y}_{>n}|\lambda[n+1]=q)/\Pr(\boldsymbol{y})\end{aligned}\tag{B.9}$$

We can now rewrite (B.9) in terms of the previously defined α, β, and γ.

$$\begin{aligned}\Pr(\lambda[n]=p,\lambda[n+1]=q|\boldsymbol{y})=&\alpha_n(p)\times\\&\gamma_n(p,q)\times\\&\beta_{n+1}(q)/\Pr(\boldsymbol{y})\end{aligned}\tag{B.10}$$

There is a specific input associated with each state transition, but there may be multiple state transitions associated with each possible input. In other words, there is a many-to-one mapping of state transitions to inputs and a one-to-many mapping of inputs to state transitions. In order to find $\Pr(a=a_x|\boldsymbol{y})$ we must sum over all the state transitions associated with the specific input a_x.

$$\Pr(a=a_x|\boldsymbol{y})=\sum_{(p,q)\in\mathcal{S}_x}\alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})\tag{B.11}$$

where $\mathcal{S}_x$ is the set of transitions associated with the specific input a_x

We can remove the $\Pr(\boldsymbol{y})$ term by recognizing summing over all $\Pr(a = a_x|\boldsymbol{y})$ for all $a_x \in \mathcal{A}$ will equal 1.

$$\sum_{a_x \in \mathcal{A}} \Pr(a = a_x|\boldsymbol{y}) = 1; \tag{B.12}$$

We can now rewrite Equation (B.11) as

$$\begin{aligned} \Pr(a = a_x|\boldsymbol{y}) &= \frac{\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})}{\sum_{a_x\in\mathcal{A}}\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})} \\ &= \frac{\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)}{\sum_{a_x\in\mathcal{A}}\sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)} \end{aligned} \tag{B.13}$$

where the denominator normalizes the probabilities to one.

Both α and β can be calculated recursively. The derivation below of the recursive calculation of α and β follows [109]. We start by writing out the definition of α_n (B.3) at the next time step.

$$\alpha_{n+1}(q) = \Pr(\lambda[n+1] = q, \boldsymbol{y}_{<n}, y_n) \tag{B.14}$$

the marginal distribution is computed from the joint, $\Pr(A) = \sum_{B\in\mathcal{B}} \Pr(A, B)$

$$\alpha_{n+1}(q) = \sum_{p=0}^{Q-1} \Pr(\lambda[n+1] = q, \boldsymbol{y}_{<n}, y_n, \lambda[n] = p) \tag{B.15}$$

where Q is the number of possible states at time n. This is a sum over all possible previous states.

We now factor the right hand side of Equation B.15 using the identity $\Pr(A, B) = \Pr(A)\Pr(B|A)$ to get

$$\alpha_{n+1}(q) = \sum_{p=0}^{Q-1} \Pr(\lambda[n] = p, \boldsymbol{y}_{<n}) \Pr(\lambda[n+1] = q, y_n|\lambda[n] = p, \boldsymbol{y}_{<n}) \tag{B.16}$$

From the Markov property if we know the state at time n, knowledge of values of $\boldsymbol{y}$ before time n are irrelevant.

$$\begin{aligned}\alpha_{n+1}(q) &= \sum_{p=0}^{Q-1} \underbrace{\Pr(\lambda[n] = p, \boldsymbol{y}_{<n})}_{\alpha_n(p)} \underbrace{\Pr(\lambda[n+1] = q, y_n | \lambda[n] = p)}_{\gamma_n(p,q)} \\ &= \sum_{p=0}^{Q-1} \alpha_n(p)\gamma_n(p,q)\end{aligned} \tag{B.17}$$

An almost identical method can be used to find a *backward* recursion for determining β resulting in

$$\begin{aligned}\beta_n(p) &= \sum_{q=0}^{Q-1} \underbrace{\Pr(\lambda[n+1] = q, y_n | \lambda[n] = p)}_{\gamma_n(p,q)} \underbrace{\Pr(\boldsymbol{y}_{>n} | \lambda[n+1] = q)}_{\beta_{n+1}(q)} \\ &= \sum_{q=0}^{Q-1} \gamma_n(p,q)\beta_{n+1}(q)\end{aligned} \tag{B.18}$$

where Q is the number of states in the next state at time $n+1$.

The term γ represents the probability of transition from p to q at time n with the observation y_n. We can factor γ

$$\begin{aligned}\gamma_n(p,q) &= \Pr(\lambda[n+1] = q, y_n | \lambda[n] = p) \\ &= \Pr(y_n | \lambda[n+1] = q, \lambda[n] = p) \underbrace{\Pr(\lambda[n+1] = q | \lambda[n] = p)}_{\textit{a priori} \text{ probabilities}}\end{aligned} \tag{B.19}$$

The second factor is the *a priori* probability $\Pr(a_x = a)$. The first factor can be rewritten as $\Pr(y_n | s_{p,q})$ where $s_{p,q}$ is the output of the channel uniquely associated with a state transition from p to q. For an AWGN channel we can write this as

$$\Pr(y_n | s_{p,q}) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(\frac{-\|y_n - s_{p,q}\|^2}{2\sigma^2}\right) \tag{B.20}$$

where σ^2 is the variance of the noise term from Equation B.1.

Table B.1: MAP Decoding Algorithm

```
$\alpha_0 \Leftarrow$ initial $\alpha$
$\beta_N \Leftarrow$ initial $\beta$
for $n = 0$ to $N-2$ do
   for $q = 0$ to $Q-1$ do
      $\alpha_{n+1}(q) \Leftarrow N_q \sum_{p=0}^{Q-1} \alpha_n(p)\gamma_n(p,q)$                {Forward}
   end for
end for
for $n = N-2$ to $0$ do
   for $p = 0$ to $Q-1$ do
      $\beta_n(p) \Leftarrow N_p \sum_{q=0}^{Q-1} \gamma_n(p,q)\beta_{n+1}(q)$                {Backward}
   end for
end for
for $n = 0$ to $N-2$ do
   for all $a_x \in \mathcal{A}$ do
      $\Pr(a = a_x|\boldsymbol{y}) \Leftarrow N_{a_x} \sum_{(p,q)\in\mathcal{S}_x} \alpha_n(p)\gamma_n(p,q)\beta_{n+1}(q)/\Pr(\boldsymbol{y})$      {a posteriori}
   end for
end for
```

For numerical stability reasons, it is common to normalize both α and β at every stage. The effect of the normalization cancels out when calculating the *a posteriori* probabilities with Equation (B.13)

A summary of the MAP algorithm for calculating the *a posteriori* probabilities is given in Table B.1. The N_x factors in front of the summations are normalizations for each step such that the αs and βs sum to one for each stage.

In summary the MAP algorithm is an efficient method for calculating the *a posteriori* probabilities. Observe from Algorithm B.1 that the calculation of the *a posteriori* probabilities involves three passes, each pass with roughly $N \times Q^2$ operations.

LIST OF REFERENCES

[1] "Technical characteristics for a universal shipborne automatic identification system using time division multiple access in the VHF maritime mobile band," 2001. ITU-R M.1371-1.

[2] S. Verdú, *Multiuser Detection.* New York, NY: Cambridge University Press, first ed., 1998.

[3] H. L. V. Trees, *Detection, Estimation, and Modulation Theory Part I.* New York, NY: Wiley, 2001.

[4] "IALA technical clarifications on ITU recommendation itu-r m.1371-1," 2003. ITU-R M.1371-1.

[5] J. Laster and J. Reed, "Interference rejection in digital wireless communications," *Signal Processing Magazine*, pp. 37–62, May 1997.

[6] H. L. V. Trees, *Optimum Array Processing.* New York, NY: Wiley, 2002.

[7] P. A. Hoeher, S. Badri-hoehr, W. Xu, and C. Krakowski, "Single-antenna co-channel interference cancellation for TDMA cellular radio systems," *IEEE Wireless Communications*, vol. 12, pp. 30–37, April 2005.

[8] J. G. Proakis, *Digital Communications.* New York, NY: McGraw-Hill, forth ed., 2001.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley and Sons, 1991.

[10] C. Huffine, "Tacsat2 cdr." Presentation at the TACSAT2 CDR, 2005.

[11] A. Papoulis, *Probability Random Variables, and Stochastic Processes.* New York: McGraw-Hill, third ed., 1991.

[12] S. Lin and D. J. C. Jr., *Error Control Coding.* Upper Saddle River, NJ: Pearson Prentice Hall, second ed., 2004.

[13] V. Franz and A. J. B., “Reduced search BCJR algorithms,” in *Int. Symp. on Information Theory*, p. 230, IEEE, June 1997.

[14] V. Franz and J. B. Anderson, “Concatenated decoding with a reduced-search BCJR algorithm,” *IEEE Journal on Selected Areas In Communications*, vol. 16, pp. 186–195, Febuary 1998.

[15] T. Eriksen, G. K. Hoye, B. Meland, B. Narheim, and B. Jenkslokken, “Maritime trafic monitoring using a space-based AIS receiver,” in *55th International Astronatical Congress 2004*, (Vancouver, Canada), pp. 1–14, October 2004.

[16] G. K. Hoye, T. Eriksen, B. Meland, and B. Narheim, “Space-based AIS for global maritime traffic monitoring,” in *5th International Symposium on Small Satellites for Earth Observation*, IAA, April 2005.

[17] T. Wahl, G. K. Høye, A. Lyngvi, and B. T. Narheim, “New possible roles of small satellites in maritime surveillance,” *Acta Astronautica*, vol. 59, no. 1-2, pp. 273–277, 2005.

[18] A. Norris, “Automatic identification systems - the effects of class b on the use of class a systems,” *Journal of Navigation*, vol. 59, pp. 335–347, April 2006.

[19] K. Murota, “GMSK modulation for digital mobile radio telephony,” *Communications, IEEE Transactions on*, vol. 29, pp. 1044–1050, July 1981.

[20] J. D. Laster, *Robust GMSK Demodulation Using Demodulator Diversity and BER Estimation.* Doctor of philosophy, Virginia Polytechnic Institude and State University, 1997.

[21] P. A. Laurent, “Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (amp),” *Communications, IEEE Transactions on*, vol. 34, pp. 150–160, Feburary 1986.

[22] K. Huang, “Supplementary proof for “exact and approximage construction of digital phase modulations by superposition of AMP” by p. a. laurent,” *Communications, IEEE Transactions on*, vol. 53, pp. 234–237, February 2005.

[23] P. Jung, "Laurent's representation of binary digital continous phase modulated signals with modulation index 1/2 revisited," *Communications, IEEE Transactions on*, vol. 42, pp. 221–224, Febuary 1994.

[24] Ö. Dural and J. G. Proakis, "The performance of simplified maximum-likelihood sequence detector for continuous phase modulation scheme," *Communications, 2000. ICC '00. IEEE International Conference on*, vol. 2, pp. 641–645, June 2000.

[25] Ö. Dural and J. G. Proakis, "Comparison of two reduced complexity maximum-likelihood sequence detectors for continuous phase modulation scheme," *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, vol. 4, pp. 2510–2513, October 2001.

[26] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal - part 2: Algorithms and applications," *IEEE Proceedings*, vol. 80, pp. 539–568, 1992.

[27] S. Kay, *Modern Spectral Estimation.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[28] S. Tretter, "Estimating the frequency of a noisy sinusoid by linear regression," *Information Theory, IEEE Transactions on*, vol. IT-31, pp. 832–835, November 1985.

[29] J. Hansen, "Selected approaches to estimation of signal phase." [online] hansenj@ele.uri.edu, 2005.

[30] B. A. Wilson, R. New, J. Campello, and B. Marcus, "Joint estimation of data and timing in the presence of inter-symbol interference," *IEEE Transactions on Magnetics*, vol. 39, pp. 2582–2584, September 2003.

[31] M. Rice, B. McIntire, and O. Haddadin, "Data-aided carrier phase estimation for GMSK," *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 5, pp. 3560–3564, May 2003.

[32] U. Mengali and A. D'Andrea, *Synchronization Techniques for Digital Receivers.* New York, NY: Plenum Press, 1997.

[33] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing," *IEEE Journal on Selected Areas In Communications*, vol. 19, pp. 2346–2357, Dec 2001.

[34] J. Huber and W. Liu, “Data-aided synchronization of coherent CPM-receivers,” *Communications, IEEE Transactions on*, vol. 40, pp. 178–189, January 1992.

[35] Y. Jiang, F.-W. Sun, and J. S. Baras, “On the performance limits of data-aided synchronization,” *Information Theory, IEEE Transactions on*, vol. IT-49, pp. 191–203, January 2003.

[36] D. J. Reader and W. J. Cowley, “Blind maximum likelihood sequence detection,” in *Acoustics, Speech and Signal Processing, 1996 ICASSP-96*, vol. 5, pp. 2694–2697, IEEE, 1996.

[37] K. M. Chugg, “Blind acquisition characteristics of PSP-based sequence detectors,” *IEEE Journal on Selected Areas In Communications*, vol. 16, pp. 1518– 1529, October 1998.

[38] J. Anderson, T. Aulin, and C. Sundberg, *Digital Phase Modulation.* New York: Plenum, 1986.

[39] Y. Huang, K. Fan, and C. Huang, “A fully digital noncoherent and coherent GMSK receiver architecture with joint symbol timing error and frequency offset estimation,” *Vehicular Technology, IEEE Transactions on*, vol. 49, pp. 863–873, May 2000.

[40] S. Ennoubi, “Comments on ”gmsk with differential phase in the mobile channel”,” *Communications, IEEE Transactions on*, vol. 40, pp. 666–669, April 1992.

[41] P. Varshnew, J. E. Salt, and S. Kumar, “BER analysis of GMSK with differential detection in a land mobile channel,” *Vehicular Technology, IEEE Transactions on*, vol. 42, pp. 683–689, November 1993.

[42] S. M. Elnoubi, “Analysis of GMSK with two-bit differential detection in land mobile radio channels,” *Communications, IEEE Transactions on*, vol. 35, pp. 237–240, Febuary 1987.

[43] S. L. Miller and R. J. O’Dea, “Multiple symbol noncoherent detection of GMSK,” *Communications, 1998. ICC ’00. IEEE International Conference on*, vol. 2, pp. 1676–1680, June 1998.

[44] G. K. Kaleh, “Simple coherent receivers for partial response continuous phase modulation,” *IEEE Journal on Selected Areas In Communications*, vol. 7, pp. 1427–1436, Dec 1989.

[45] N. Al-Dhahir and G. Saulnier, “A high-performance reduced-complexity GMSK demodulator,” *Communications, IEEE Transactions on*, vol. 46, pp. 1409–1413, November 1998.

[46] A. Y. glu, D. Markrakis, and K. Feher, “Differential detection of GMSK using decision feedback,” *Communications, IEEE Transactions on*, vol. 36, pp. 641–649, June 1988.

[47] S. K. Yao, J. H. Reed, and J. D. Laster, “GMSK differential detectors with decision feedback in multipath CCI channels,” *Communications, IEEE Transactions on*, vol. 36, pp. 1830–1834, June 1988.

[48] A. Abrardo, G. Benelli, and G. R. Cau, “Multiple-symbol differential detection of GMSK for mobile communications,” *Vehicular Technology, IEEE Transactions on*, vol. 44, pp. 379–389, August 1995.

[49] B. A. Bjerk, J. G. Proakis, K. Y. M. Lee, and Z. Zvonar, “A comparison of GSM receivers for fading multipath channels with adjacent- and co-channel interference,” *IEEE Journal on Selected Areas In Communications*, vol. 18, pp. 2211– 2219, November 2000.

[50] B. Sklar, “How i learned to love the trellis,” *Signal Processing Magazine*, vol. 20, pp. 87–102, May 2003.

[51] G. D. Forney, “Maximum-likelihood sequence estimation of digital sequences in the presense of intersymbol interference,” *Information Theory, IEEE Transactions on*, vol. IT-18, pp. 363–378, May 1972.

[52] E. A. Lee and D. G. Messerschmitt, *Digital Communications.* Boston: Kluwer Academic Publishers, second ed., 1994.

[53] S. Verdú, “Maximum likelihood sequence detection for intersymbol interference channels: A new upper bound on error probability,” *Information Theory, IEEE Transactions on*, vol. IT-33, pp. 62–68, January 1987.

[54] G. Ungerboeck, “Adaptive maximum-likelihood receiver for carrier-modulated data transmission systems,” *Communications, IEEE Transactions on*, vol. 22, pp. 624–636, May 1974.

[55] G. E. Bottomley and S. Chennakeshu, “Unification of MLSE receivers and extension to time-varying channels,” *Communications, IEEE Transactions on*, vol. 46, pp. 464–472, April 1998.

[56] D. A. Shnidman, “A generalized Nyquist criterion and optimum linear reciever for a pulse modulation system,” *Bell Systems Technical Journal*, vol. 46, pp. 2163–2177, November 1967.

[57] A. R. Kaye and D. A. George, “Transmission of multiplexed PAM signals over multiple channel and diversity systems,” *Communications, IEEE Transactions on*, vol. 18, pp. 520–526, October 1970.

[58] L. Nie, “Comparison of exact and approximate multi-user detection for GSM,” Master’s thesis, Informatics and Mathematical Modelling Technical University of Denmark, 2005.

[59] W. van Etten, “Maximum likelihood receiver for multiple channel transmission systems,” *Communications, IEEE Transactions on*, vol. 24, pp. 276–283, Febuary 1976.

[60] H. Trigui and D. T. M. Slock, “Training sequence aided multichannel identification in the presense of interference and noise,” in *Global Telecomunications Conference Globecom*, pp. 159–164, IEEE, 1998.

[61] H. Trigui and D. T. M. Slock, “Cochannel interference cancellation within the current GSM standard,” in *Universal Personal Communications*, pp. 511–516, IEEE, October 1998.

[62] H. Trigui and D. T. M. Slock, “Training sequence based multiuser channel identification for cochannel interference cancellation in GSM,” in *Signal Processing Advances in Wireless Communications*, pp. 102–105, IEEE, 1999.

[63] R. Lópex-Valcarce, Z. Ding, and S. Dagupta, “Equalization and interference cancellation in linear multiuser systems based on second-order statistics,” *Signal Processing, IEEE Transactions on*, vol. 49, pp. 2042–2049, September 2001.

[64] A. Batra, *Extensions of the Constant-Modulas Algorithm and the Phase-Locked Loop for Blind Multiuser Detection.* Doctor of philosophy, Georgia Institute of Technology, 2000.

[65] L. B. Lopes, "Coherent and limiter-discriminator detection of GMSK in interference limited conditions," *Electronics Letters*, vol. 27, pp. 2313–2315, December 1995.

[66] M. Pukkila, G. P. Mattellini, and R. P.A., "Constant modulas single antenna interference cancellation for GSM," *Vehicular Technology Conference, 2004. VTC 2004 Spring. IEEE VTS 59th*, vol. 1, pp. 584–588, May 2004.

[67] M. Pukkila, G. P. Mattellini, and R. P.A., "Cochannel interference suppression for constant modulas signals," *Communications, IEEE Transactions on*, vol. 5, pp. 2548–2552, June 2004.

[68] P. A. Murphy, *Optimum and Reduced Complexity Multiuser Detectos for Synchronous and Asynchronous CPM Signaling.* Doctor of philosophy, University of California at Davis, 1998.

[69] P. A. Murphy and G. E. Ford, "Co-channel demodulation for continous-phase modulated signals," in *ASILOMAR*, pp. 330–334, IEEE, 1996.

[70] P. A. Murphy, M. Golanbari, and G. E. Ford, "A generalized multiuser detector for asynchrounous partial-response PAM signals," in *ASILOMAR*, pp. 1210–1214, IEEE, 2001.

[71] A. Polydoros and K. Chugg, "Co-channel interference mitigation techniques for narrowband signals," *X National Symposium of Radio Science, U.R.S.I.*, March 2001.

[72] H. Schoeneich and P. A. Hoeher, "Bit error bound of maximum-likelihood cochannel interference cancellation," 2003.

[73] S. Verdú, "Minimum probability of error for asynchronous gaussian multiple-access channels," *Information Theory, IEEE Transactions on*, vol. IT-32, pp. 85–96, January 1986.

[74] K. Giridhar, A. Mathur, and J. J. Shynk, "A blind adaptive MAP algorithm for the recovery of cochannel signals," in *Military Communications Conference*, vol. 1, pp. 133–138, IEEE, 1994.

[75] K. Abend and B. D. Fritchman, “Statistical detection for communications channels with intersymbol interference,” *IEEE Proceedings*, vol. 58, pp. 779–785, May 1970.

[76] K. Giridhar, J. J. Shynk, A. Mathur, S. Chari, and R. P. Gooch, “Nonlinear techniques for the joint estimation of cochannel signals,” *Communications, IEEE Transactions on*, vol. 45, pp. 473–484, April 1997.

[77] H. Arslan, S. C. Gupta, G. E. Bottomley, and S. Chennakashu, “New approaches to adjacent channel interference suppression in FDMA/TDMA mobile radio systems,” *Vehicular Technology, IEEE Transactions on*, vol. 49, pp. 1126–1139, July 2000.

[78] P. Barthelemy, “A model-based receiver for CPM signals in a cochannel interference limited environment,” Master’s thesis, Virginia Polytechnic Institute and State University, 2002.

[79] F. Xiong and E. Shwedyk, “Sequential sequence estimation for multiple-channel systems with intersymbol and interchannel interference,” *Communications, IEEE Transactions on*, vol. 41, pp. 322–331, Febuary 1993.

[80] Y. C. Eldar and A. M. Chan, “An optimum whitening appoach to linear multiuser detection,” *Information Theory, IEEE Transactions on*, vol. IT-49, pp. 2156–2171, September 2003.

[81] P. A. Ranta, A. Hottinen, and Z.-C. Honkasalo, “Co-channel interference cancelling receiver for TDMA mobile systems,” *Communications, 1995. ICC ’95. IEEE International Conference on*, vol. 1, pp. 17–21, June 1995.

[82] W. van Etten, “An optimum linear receiver for multiple channel digital transmission,” *Communications, IEEE Transactions on*, vol. 23, pp. 828–834, August 1975.

[83] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *Information Theory, IEEE Transactions on*, vol. 20, pp. 284–287, March 1974.

[84] Y. Li, B. Vecetic, and Y. Sato, “Optimum soft-output detection for channels with intersymbol interference,” *IEEE Transactions on Information Theory*, vol. 41, pp. 704–713, May 1995.

[85] M. El-Assal and M. Bayoumi, “A hight speed architecture for MAP decoder,” *Signal Processing Systems , IEEE workshop on*, pp. 69–74, October 2002.

[86] G. Colavolpe, G. Ferrari, and R. Raheli, “Reduced-state BCJR-type algorithms,” *IEEE Journal on Selected Areas In Communications*, vol. 19, pp. 848– 859, May 2001.

[87] C. M. Vithanage, C. Angrieu, and R. J. Peichocki, “Approximate inference in hidden markov models using iterative active state selection,” *Signal Processing Letters, IEEE*, vol. 13, pp. 65–68, Febuary 2006.

[88] “Adoption of amendements to the international convention for the safety of life at sea, 1974, as amended,” 2000. Resolution MSC.99(73).

[89] “Telecommunications and information exchange between systems – high-level data link control (hdlc) procedures frame structure,” 1993. ISO/IEC 3309.

[90] J. Makhoul, “Linear prediction: A tutorial review,” *IEEE Proceedings*, vol. 63, pp. 561–580, April 1975.

[91] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* New York: John Wiley and Sons, third ed., 1997.

[92] L. J. Levy, “The kalman filter: Navigation’s integration workhorse,” *GPS World*, pp. 65–71, September 1997.

[93] D. Bertsckas and R. G. Gallager, *Data Networks.* Upper Saddle River, NJ: Prentice Hall, second ed., 1987.

[94] G. Kabatjanskii, A. J. H. Vinck, and A. J. van Wijngaarden, “On combined synchronization and error control coding,” in *Int. Symp. on Information Theory*, p. 62, IEEE, June 1994.

[95] A. J. van Wijngaarden and K. A. S. Immink, “Construction of constrained codes using sequence replacement techniques,” *IEEE Journal on Selected Areas In Communications*, vol. 19, pp. 602–611, April 2001.

[96] M. T., K. F., and H. J., “Maximum likelihood sequence detection of a bit-stuffed data source,” *Communications, 2006. ICC ’06. IEEE International Conference on*, June 2006.

[97] T. Nolte, H. Hansson, and C. Norstrom, “Using bit-stuffing distributions in can analysis,” in *IEEE/IEE Real-Time Embedded Systems Workshop*, Dec 2001.

[98] J. C. Fu and W. Y. W. Lou, *Distribution Theory of Runs and Patterns and its Applications*. Singapore: World Scientific Publishing Co. Pte. Ltd., first ed., 2003.

[99] A. N. Philippou and F. S. Makri, “Success runs and longest runs,” *Statistics and Probability Letters*, vol. 4, pp. 211–215, 1986.

[100] K. Hirano, “Some properties of the distributions of order k,” in *Fibonacci Numbers and Their Applications*, (Reidel, Dordrecht), pp. 43–53, 1986.

[101] J. L. Melsa and D. L. Cohn, *Decision and Estimation Theory*. New York: McGraw-Hill, first ed., 1978.

[102] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, second ed., 2001.

[103] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[104] G. Colabolpe and A. Barbieri, “On MAP symbol detection for ISI channels using the Ungerboeck observation model,” *Communications Letters, IEEE*, vol. 9, pp. 720–722, August 2005.

[105] S. J. Leon, *Linear Algebra with Applications*. New York: Macmillan, third ed., 1990.

[106] M. C. Valenti, *Iterative Detection and Decoding for Wireless Communications*. Doctor of philosophy, Virginia Polytechnic Institute and State University, 1999.

[107] M. Brookes, “The matrix reference manual.” [online] http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html, 2005.

[108] J. Hicks, “Symbol-step mapse of complex pam signals received over known isi channels with field-oriented a priori information: Trellis structure and simplifications.” unpublished write up by hicks based on discussion with meehan, November 2005.

[109] T. K. Moon, *Error Correction Coding*. Hoboken, New Jersey: Wiley, first ed., 2005.

[110] W. J. Larson and J. R. Wertz, *Space Mission Analysis and Design.* Torrance, California: Microcosm, Inc, second ed., 1992.

[111] P. Z. P. Jr, *Probability Random Variables, and Random Signal Principles.* New York: McGraw-Hill, forth ed., 2001.

[112] A. E. Gamal and T. M. Cover, “Multiple user information theory,” *IEEE Proceedings*, vol. 68, pp. 1466–1483, December 1980.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Dept. of Electrical and Computer Engineering
   Attn: Prof. Jeffrey B. Knorr, Code EC/Ko
   Naval Postgraduate School
   Monterey, California

4. Prof. Frank E. Kragh, Code EC/Kh
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Prof. Herschel H. Loomis, Code EC/Lm
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Prof. Alan A. Ross, Code SP/Ra
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

7. Prof. Charles W. Therrien, Code EC/Ti
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

8. Dr. Kenneth A. Clark, Code 8121
   Naval Center for Space Technology
   Naval Research Laboratory
   Washington, DC

9. Prof. Murali Tummala, Code EC/Tu
   Department of Electrical and Computer Engineering

Naval Postgraduate School
Monterey, California

10. Timothy J. Meehan
Naval Postgraduate School
Monterey, California